

Cuda 12000 IP Access Switch CLI Reference Guide

Release 3.0

PART NO. 780-000049-00
PUBLISHED SEPTEMBER 2001

http://www.adc.com

ADC Telecommunications, Inc.
8 Technology Drive
Westborough, MA 01581

ADC Telecommunications, Inc. (herein referred to as "ADC") may revise this manual at any time without notice. All statements, technical information, and recommendations contained herein are believed to be accurate and reliable at the time of publication but are presented without any warranty of any kind, express or implied, (including the warranties of merchantability and fitness and against infringement or interferrence with your enjoyment of the information) and you are solely responsible for your use of this manual with any equipment or software described herein.

This manual (in whole or in part, including all files, data, documentation, and digital and printed copies made therefrom) is protected by United States copyright laws, international treaties and all other applicable national or international laws. With the exception of materials printed for use by a user who is authorized by separate license from ADC, this manual may not, in whole or part, be modified, excerpted, copied, photocopied, translated, or reduced to any electronic medium or machine readable form, without ADC's written consent obtained prior thereto.

The CUDA 12000 is listed to UL 1950 Third Edition and CAN/CSA-C22.2 No. 950-95 Third Edition compliance.

The following information is for compliance by Class A devices with FCC regulations: the equipment described in this manual has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC regulations. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case you will be required to correct the interference at your own expense.

You can determine whether your equipment is causing interference by turning it off. If the interference stops, it was probably caused by the equipment or one of its peripheral devices. If the equipment causes interference to radio or television reception, try to correct the interference by using one or more of the following methods.

- Turn television or radio antenna until the interference stops.
- Move equipment to one side or the other of the television or radio.
- Move equipment farther away from the television or radio.
- Plug equipment into an outlet that is on a different circuit from the television or radio. *(That is, make certain the equipment and the television or radio are on circuits controlled by different circuit breakers or fuses.)*

Modifications to this equipment that are not authorized by ADC could void the FCC certification and UL approval and negate your authority to operate the equipment.

This manual is provided by ADC on an "AS IS, WITH ALL FAULTS" basis, without any representation or warranty of any kind, either express or implied, including without limitation any representations or endorsements regarding use of, the results of, or performance of the equipment or software, its appropriateness, accuracy, reliability, or correctness. **You assume the entire risk as to the use of this manual. ADC does not assume liability for the use of this manual beyond its original purchase price. In no event will ADC be liable for additional direct or indirect damages including any lost profits, lost savings, lost revenue or other incidental or consequential damages arising from any defects, or the use or inability to use this manual or the equipment or software described herein, even if ADC has been advised of the possibility of such damages.**

Cuda 12000, MeshFlow, CudaView, FastFlow Broadband Provisioning Manager and CableOnce are trademarks of ADC Telecommunications, Inc. CableLabs® is a registered trademark of Cable Television Laboratories, Inc. Java® is a registered trademark of Sun Microsystems, Inc. in the United States and other countries. Jini™ is a trademark of Sun Microsystems, Inc. in the United States and other countries.

The Cuda 12000 includes RSA BSAFE cryptographic or security protocol software from RSA security. The Cuda 12000 contains an integrated DOCSIS-compliant provisioning server. Use of this provisioning functionality is restricted to licensed authorization. ADC will not support provisioning for for your use thereof if you are not authorized by the appropriate software license to use such provisioning.

All other company and product names mentioned herein may be trademarks of their respective companies.

The equipment and software described herein may be covered by an ADC warranty statement. You may obtain a copy of the applicable warranty by referring to www.adc.com/cable/support and selecting the *technical assistance* link. What follows is a summary of the warranty statement. The summary is not binding on ADC and is provided to you merely as a convenience.

The equipment warranty usually lasts twelve (12) months from point of shipment and the software warranty usually lasts sixty (60) days from the point of shipment. The software warranty covers both functionality as well as the media on which the software is delivered. Neither warranty entitles the customer to receive free and unlimited access for technical assistance. A separate technical support agreement must be purchased for unlimited access to technical support resources.

The equipment warranty only applies to the cost of a replacement component. It does not include the labor charge for installation of the replacement component. During the warranty period, warranty claims will be processed on a 10-day return to factory basis. Once the defective component is returned to the factory, ADC's sole liability under the equipment warranty shall be either:

- To repair or to replace, at ADC's option, the defective equipment component with a new or refurbished component; or
- If after repeated efforts ADC is unable to resolve the defect by repair or replacement, to refund the purchase price of the equipment or component upon return of the defective item.

A working component will be returned to the customer within 10 days after it is received by ADC.

The warranty period for repaired or replaced equipment components shall be the remainder of the original warranty period for the repaired or replaced item or ninety (90) days, whichever is greater.

Equipment warranty claims can be processed on-line through a web interface or directly by a customer support representative of ADC. As part of the standard process for issuing a Return Materials Authorization (RMA), the Customer Support organization will verify all reported failures prior to authorizing a shipment of a replacement part.

The equipment warranty does not cover any of the following events:

- The equipment has been subject to abnormal use, abnormal conditions, improper storage, exposure to moisture or dampness, unauthorized modifications, unauthorized connections, unauthorized repair, misuse, neglect, abuse, accident, alteration, improper installation, or other events which are not the fault of ADC, including damage caused by shipping;
- ADC or an authorized ADC distributor or reseller was not notified by the customer of the equipment defect during the applicable warranty period.

If the software media is unusable such that the software cannot be loaded onto the equipment, ADC will replace the media within 1 business day after ADC is notified through Technical Assistance Center.

During the software warranty period, ADC will provide software updates and/or maintenance releases at no additional charge to resolve any issues where the software does not function according to software specification. In order to receive on-going software maintenance releases after the 60-day warranty period, the customer must purchase the base level technical assistance agreement.

The software warranty does not cover any of the following events:

- Unauthorized modifications to the software or firmware;
- Unauthorized installation of non-ADC software on the Cuda 12000 platform;
- ADC or an authorized ADC distributor or reseller was not notified by the customer of the software defect during the applicable warranty period.

Non-ADC software may be warranted by its developer, owner or other authorized entity as expressly provided in the documentation accompanying such software.

Failures caused by non-ADC software are not covered by ADC's warranty and service activities to remedy such failures will be billed to the customer.

Remote technical assistance will be provided free of charge during the 60-day software warranty period. The hours for support during the warranty period are Monday through Friday from 8:00am to 5:00pm EST.

Additional hardware and software services are available by purchasing an extended service agreement. Contact your account representative or call 1-877-227-9783 for further details.

© 2001 ADC Telecommunications, Inc. All Rights Reserved.

CONTENTS

CUDA 12000 IP ACCESS SWITCH CLI REFERENCE GUIDE

ABOUT THIS GUIDE

Document Objective 14
Audience 14
Document Organization 15
Notations 16
Command Syntax 17
Related Documentation 18
Contacting Customer Support 19

1 ABOUT THE COMMAND LINE INTERFACE

About the CLI 21
Accessing the CLI 23
Command Modes 26
- Global Commands 28
- Root Mode 30
- Physical Interface Mode 32
- IP Interface Mode 36
- OSPF Global Configuration Mode 37
- Import and Export OSPF Route Filter Modes 39
- RIP Configuration Mode 40
- Import and Export RIP Route Filter Modes 41
- Slot Mode 42

2 CUDA 12000 BASE SYSTEM COMMANDS

aaa authentication 44
access-class 45
access-list 47
add arp 50
admission-control 51
alarm-throttle 52
arp timeout 53
asbr 54
aux-device ac-monitor 55
aux-device backplane-clock-a 56
aux-device backplane-clock-b 57
aux-device db15 alarm 58
aux-device dc-monitor 62
aux-device fan-rotation 63
aux-device fan-temp 64
aux-device ps-temp 65
basmonitor 66
boot 67
bootp-policy 68
bridge-group 70
bridge-interface 72
bridge-timeout 74
ccdown 76
chassis 77
chassis-config 78
chassis-fault 80
clear arp-cache 83
clear counters 84
clear ip igmp group 85
clear service-flow log 86
clock-source 87
cm-filter 88
cm-filter-default cm downstream 92
cm-filter-default cm upstream 94
cm-filter-default cpe downstream 96

cm-filter-default cpe upstream 98
cm-offline clear 100
cm-offline persist 102
cm-offline timer 103
cm cpe-reset 104
cm modify active 109
cm modify cm-downstream 111
cm modify cm-upstream 113
cm modify cpe-downstream 115
cm modify cpe-upstream 117
cm modify learnable 119
cm modify max-ip 121
cm modify upstream 123
cm reset 125
connect 129
cpe-control active 130
cpe-control learnable 131
cpe-control max-ip 132
cpu-utilization 133
crc 134
db-check 135
db-connect 136
del arp 138
dhcp-authority 139
dhcp-policy 142
dhcp-relay 145
downstream frequency 148
downstream interleave-depth 150
downstream modulation 151
downstream no shutdown 153
downstream shutdown 154
downstream transmit-power 155
duplex 156
enable 158
event-config reporting 159
event-config syslog 162
event-config throttle 163
event-log clear 165

export 166
filter-aging 167
flap-list aging 169
flap-list clear 170
flap-list insertion-time 172
flap-list power-adj-threshold 174
flap-list size 175
help 176
http-server 178
import 179
insertion-interval 180
interface 181
ip address 183
ip filter 185
ip igmp 187
ip igmp proxy 189
ip ospf area-id 191
ip ospf authentication 193
ip ospf cost 196
ip ospf dead-interval 199
ip ospf hello-interval 202
ip ospf priority 205
ip ospf retransmit-interval 208
ip ospf transit-delay 211
ip rip accept default-route 214
ip rip accept host-route 216
ip rip authentication 218
ip rip cost 221
ip rip default cost 223
ip rip disable 225
ip rip enable 227
ip rip neighbor 229
ip rip poisoned-reverse 231
ip rip receive-version 233
ip rip send-version 235
ip rip send default-also 237
ip rip send default-only 239
ip rip split-horizon 241

ip route 243
ip route default 245
ip source-route 246
loop 247
link-trap 249
lookup 250
map-list 251
map-timer 253
match 254
modulation-profile 256
mtu 258
negotiation auto 259
no access-class 260
no access-list 262
no asbr 263
no bootp-policy 264
no clock-source 265
no crc 266
no db-connect 267
no dhcp-relay 268
no enable 269
no ip address 270
no ip igmp 271
no ip igmp proxy 273
no ip rip 274
no loop 275
no modulation-profile 276
no mtu 277
no ospf area 278
no pos flag 280
no pos report 281
no pos scramble 282
no pos threshold 283
no ppp authentication 284
no ppp chap-hostname 285
no ppp chap-password 286
no ppp ipcp-report-address 287
no ppp negotiation-count 288

no ppp pap-sent-username 289
no ppp timeout 290
no ppp username 291
no proxy-arp 292
no snmp-server contact 293
no snmp-server context 294
no snmp-server community 295
no snmp-server group 296
no snmp-server host 297
no snmp-server location 299
no snmp-server name 300
no snmp-server user 301
no snmp-server view 302
ospf-vi 303
ospf 306
ospf area 307
override 309
periodic-ranging-interval 311
ping 312
plant-delay 314
pll-state 315
pos flag 316
pos report 317
pos scramble 320
pos threshold 321
ppp authentication 322
ppp chap-hostname 323
ppp chap-password 324
ppp ipcp-report-address 325
ppp negotiation-count 326
ppp pap-sent-username 327
ppp username 328
privacy auth 329
privacy base auth-lifetime 331
privacy base cert-trust 332
privacy base enable-cert-validity-periods 333
privacy base tek-lifetime 334
privacy ca-cert 335

privacy cm-cert 337
privacy encryption 338
privacy multicast ip 339
privacy multicast mac 341
privacy tek 342
proxy-arp 343
qos permission 344
quit 345
radius-server 346
ranging-attempts 347
report 348
reset 350
reset rip stats 351
rip 352
root 353
route-map 354
router 356
router-id 357
router ospf 359
router rip 360
save 361
set paging 362
set password 363
set prompt 364
set time 365
set timeout 366
shared-secret 367
show aaa 368
show access-class 369
show access-list 370
show admission-control 372
show alarm-throttle 373
show arp 374
show arp timeout 375
show aux-device 376
show boot 378
show bootp-policy 379
show bridge-group 381

show bridge-timeout 383
show buffers 384
show chassis 385
show chassis-config 386
show chassis-fault status 387
show clock-synchronization 388
show cm-filter 389
show cm-filter-default 390
show cm-offline 391
show controllers 392
show cpe-control 395
show cpu-utilization 396
show db-connect 398
show dhcp-authority 399
show dhcp-policy 401
show dhcp-relay 402
show downstream 403
show event-config 404
show event-log 407
show filter-aging 409
show flap-list 410
show flash 411
show host 412
show in-counters 413
show interface 414
show interface cable <c/s/i> cm-filter-default 415
show interface cable <c/s/i> cpe-control 416
show interface cable <c/s/i> downstream 417
show interface cable <c/s/i> dynamic-service-stats 418
show interface cable <c/s/i> flap-list 420
show interface cable <c/s/i> mac 421
show interface cable <c/s/i> modem 423
show interface cable <c/s/i> modem summary 424
show interface cable <c/s/i> modulation-profile 425
show interface cable <c/s/i> pll-state 426
show interface cable <c/s/i> proxy-arp 427
show interface cable <c/s/i> qos 428
show interface cable <c/s/i> sid 429

show interface cable <c/s/i> signal-quality 431
show interface cable <c/s/i> upstream 433
show interface ethernet 435
show interface pos 437
show ip 438
show ip {mroute | igmp} 439
show ip address 443
show ip filter 444
show ip interface 445
show ip interface source-route 447
show ip ospf 448
show ip rip 450
show link-trap 452
show lookup 453
show mac 454
show map-list 456
show mode 458
show modem 459
show modem <mac-address> 467
show modem summary 478
show mta <mac-address> base 479
show mta <mac-address> ncs 481
show mta <mac-address> security 485
show mta <mac-address> server 487
show ospf 489
show out-counters 491
show pll-state 492
show privacy auth 493
show privacy base 495
show privacy ca-cert 497
show privacy cm-cert 500
show privacy multicast 504
show privacy tek 506
show proxy-arp 508
show qos 509
show radius-server 510
show relay-to-syslog 511
show rip neighbors 512

show rip stats all 513
show rip stats current 515
show route-map 517
show running-config 519
show shared-secret 521
show snmp 522
show snmp community 523
show snmp context 525
show snmp engineID 526
show snmp group 527
show snmp host 528
show snmp notify 530
show snmp user 532
show snmp view 533
show spectrum-group 535
show syslog-server 538
show system 539
show tacacs-server 540
show time 541
show timeout 542
show topology 543
show trace-log 544
show traffic-relay 545
show upstream 547
show user 550
show version 551
shutdown 553
sleep 554
slot 555
snmp-server community 556
snmp-server contact 558
snmp-server context 559
snmp-server group 561
snmp-server host 564
snmp-server location 572
snmp-server name 573
snmp-server user 574
snmp-server view 576

spectrum-group 579
spectrum-group reset upstream 581
speed 582
sync-interval 584
tacacs-server 585
talk 586
trace-log 587
traceroute 593
traffic-relay 596
ucd-interval 598
up 599
upstream <port number> channel-width 600
upstream <port number> data-backoff 601
upstream <port number> frequency 602
upstream <port number> map 603
upstream <port number> minislot-size 605
upstream <port number> modulation-profile 607
upstream <port number> no shutdown 608
upstream <port number> power-level 609
upstream <port number> range-backoff 610
upstream <port number> ranging 611
upstream <port number> shutdown 613
upstream <port number> voice-bw-reserve 614

3 ACCOUNT ADMINISTRATION COMMANDS

access-profile 618
account 621
no access-profile 623
no account 624
show access-profile 625
show active-user 626
show account 627

A GLOSSARY

INDEX

ABOUT THIS GUIDE

This chapter introduces you to the *Cuda 12000 IP Access Switch CLI Reference Guide* and contains the following sections:

- Document Objective
- Audience
- Document Organization
- Notations
- Command Syntax
- Related Documentation
- Contacting Customer Support

Document Objective

The *Cuda 12000 IP Access Switch CLI Reference Guide* provides you with reference information about the commands you can use to configure and manage the Cuda 12000 system using the command line interface (CLI). Before you use this guide, you should have already installed and brought the system online using the *Cuda 12000 IP Access Switch Installation Guide*.

The *Cuda 12000 IP Access Switch CLI Reference Guide* is a companion to the The *Cuda 12000 IP Access Switch CLI-based Administration Guide*, which is a procedural guide that provides step-by-step instructions on managing the Cuda 12000 system using the CLI.

Audience

The guide targets the network administrator who is responsible for configuring and managing the Cuda 12000 within a cable television headend site. It assumes a working knowledge of network operations, although it does not assume prior knowledge of ADC's network equipment.

Document Organization

The *Cuda 12000 IP Access Switch CLI Reference Guide* is comprised of the following sections:

- **Chapter 1: About the Command Line Interface** — Introduces you to the CLI.
- **Chapter 2: Cuda 12000 Base System Commands** — Describes commands available within the Cuda 12000 IP Access Switch Base System Software.
- **Chapter 3: Account Administration Commands** — Describes commands used to configure user accounts.
- **Glossary** — Provides a glossary of terms.

Notations

This table lists the text notations are used throughout the Cuda 12000 documentation set.

Icon	Notice Type	Description
	Information Note	Important or useful information, such as features or instructions
	Caution	Information that alerts you to potential damage to the system
	Warning	Information that alerts you to potential personal injury

Command Syntax

This table lists the command syntax conventions used in this guide.

Command Element	Syntax
Commands and keywords	Expressed in **bold**. For example: **show chassis-config**
Variables	Enclosed in < > and expressed in plain text. For example: **add arp** <ip-address> <mac-address> In this example, <ip-address> and <mac-address> are variables that follow the **add arp** command.
Optional Arguments	Enclosed in [ ]. For example: **ip route** <dest-network> <mask> [**gateway**] [**metric**] In this example, the keywords **gateway** and **metric** are optional arguments.
Set of Choices	Enclosed in { | }. For example: **loop** {**line** | **internal**} In this example, the user can specify either the **line** keyword or the **internal** keyword following the **loop** command.
List	Expressed as three dots (...). For example: **snmp-server host** [<**notification-type**>...] In this example, the user can specify multiple notification types.



In examples only, all user input — commands, keywords, and variables — are in bold to distinguish what the user enters from display-only screen text. In all other sections of this document, the conventions described above apply.

Related Documentation

For more information on the Cuda 12000 system, see the following publications:

- ***Cuda 12000 IP Access Switch CLI-based Administration Guide —*** A procedural guide containing all the information you need to configure and manage the system using the Cuda 12000 CLI. The *Cuda 12000 IP Access Switch CLI-based Administration Guide* is a companion to this guide.
- ***Cuda 12000 IP Access Switch Installation Guide*** — Provides all the information you need to install the system and bring it online. Includes a test procedure to ensure that the system is operational and can provision modems.
- ***Cuda 12000 IP Access Switch CudaView Administration Guide*** — A procedural guide containing all the information you need to configure and manage the system using the CudaView graphical user interface (GUI).

Contacting Customer Support

To help you resolve any issues that you may encounter when installing, maintaining, and operating the Cuda 12000 system, you can reach ADC Customer Support as follows:

- Phone: (877) 227-9783 (option 4)
- E-mail: support@basystems.com
- Customer Support Web Site — To access Customer Support on the Web, go to `http://www.adc.com/cable/support`, then select the *Technical Assistance Center* link. You can then report the problem online, search the ADC Customer Support database for known problems and solutions, and check *Frequently Asked Questions*.

When contacting Customer Support for technical assistance, be sure to have this information ready:

- List of system hardware and software components, including revision levels and serial numbers.
- Diagnostic error messages.
- Details about recent system configuration changes, if applicable.

1 About the Command Line Interface

This chapter introduces you to the command line interface (CLI) and covers the following topics:

- About the CLI
- Accessing the CLI
- Command Modes

About the CLI

The Cuda 12000 management module runs the Linux operating system. The CLI operates within this environment. The CLI is a textual command line interface accessible through a local COM port or through remote Telnet or secure shell (SSH).

The CLI operates within the command shell and offers a number of features to facilitate ease-of-use and configuration, including:

- **Context Sensitive Online Help** — The CLI offers the following online Help mechanisms:
  - **Individual Command Help** — You can display help on most commands by typing **help** followed by the command name. For example:

    ```
    cli:172.16.19.10:root# help bridge-group
    bridge-group                Creates a bridge group
        <name/id of bridge group>
    cli:172.16.19.10:root#
    ```

    The command name is listed on the left with a description on the right. Arguments are indented in standard syntax below the command name.

  - **Command Mode Help** — To view all commands available in the current mode with associated descriptions, type **help**. To show a list of available commands without descriptions, type **?** at the prompt or press the **Tab** key twice.

- **Configurable Prompt** — By default, the prompt displays both the address assigned to the management module and the current command mode. You can configure the prompt so it does not display this information. When the address and mode is displayed in the prompt, you can issue **set prompt** to remove it. To configure the prompt to display address and mode information, issue **set prompt mode**. For example:

  ```
  cli:172.16.19.10:root# set prompt
  cli#
  cli# set prompt mode
  cli:172.16.19.10:root#
  ```

- **Command Completion** — The system does not require that you type the entire command string. You simply need to type enough of the string to make it unique among the available commands so the system can recognize it. Once you type enough of the command string to distinguish it among other commands, simply press `[Tab]` to complete the command, or press `[Enter]` to execute it.

For most commands within the CLI, hyphens are placed between nouns, (such as **cpe-control***),* while no hyphen is placed between verbs and nouns *(*such as **no shutdown** and **show ip***)*. Also note that commands and their associated arguments are case-sensitive.

Accessing the CLI

Your first form of access to the CLI (after installing the Cuda 12000) is through COM port 1 located on the front of the management module. Once you assign the Craft Ethernet port on the management module an IP address, you can access the CLI remotely through Telnet or SSH.

Use the following procedure to logon to the system management module and access the CLI environment through COM port 1:

1. Ensure that you have cabled a console or a system running a terminal emulation program to COM port 1 and configured the correct serial transmission settings (57,600, 8, 1).
2. Access the system through the terminal emulator. Press [Enter] until you see the Linux login prompt.
3. You are then prompted for a login name and password to logon to the CLI. Enter your login name and password. The system ships with the following system defaults:
   - **Account Name:** administrator
   - **Password:** bas

   The Linux prompt is then displayed.



Note that the default login name and password are case-sensitive — all lowercase.

4. From the command prompt, access the CLI environment by issuing the following command:

   bascli
5. Within the CLI environment, enter your Cuda 12000 login name and password, as follows:

   enable <account name>

   <password>

The system ships with the following system defaults:

Account Name: root

Password: bas

For example:

```
[administrator@Tech2000 administrator]$ bascli
cli:null:root> enable root
password: ***
Connecting to 172.16.19.10...
Java Server version is compatible
ClientMode: CLI
logon complete
Sending message: User root just logged in from Tech2000

FROM:root@Tech2000:: User root just logged in from
Tech2000
```



Note that the default login name and password are case-sensitive — all lowercase.

Use the following procedure to logon to the system management module and access the CLI environment through the Craft Ethernet port:

1. Ensure that you have assigned an IP address to the Ethernet craft port on the management module, and that the Telnet and SSH server processes are running.
2. Open a Telnet session or an SSH session with the IP address or hostname assigned to the management module.
3. When the cli:null:root prompt appears, enter your Cuda 12000 login name and password, as follows:

   enable <account name>

   <password>

   The system ships with the following system defaults:

   Account Name: root

   Password: bas

For example:

```
ADC Cuda 12000

cli:null:root> enable root
password: ***
Connecting to 192.168.208.3...
Java Server version is compatible
logon complete
Sending message: User root just logged in from techpubs

FROM:root@techpubs:: User root just logged in from
techpubs
```



Note that the default login name and password are case-sensitive — all lowercase.

Command Modes

The Cuda 12000 switches and routes IP traffic between cable modems on an analog HFC network, and an IP digital network. As a result, administration tasks range from configuring IP interfaces and routing protocols to managing subscribers.

To support these administration tasks, the system provides a set of global commands and multiple command modes.

Global commands can be accessed anywhere in the CLI, while each command mode provides access to a set of related commands that cover a particular configuration scope. The current command mode is displayed in the prompt by default; you can verify the current mode that you are in at anytime by using the **show mode** command.

Command mode structure follows a hierarchy in which some modes run within others; all run within root mode. You can back up to the parent level from any sub mode using the **up** command. For local access, note that you can exit the CLI command shell and return to the Linux prompt at any time by typing **quit.** For Telnet or SSH access, the **quit** command terminates your session (you can also type **q**, which is a shortened form of **quit**).

You can also display a list of all available commands within the current mode by using one of the following help commands:

Help Command	Description
help	Displays the commands and command descriptions.
?	Displays all commands available within the current mode without descriptions; you can also display all commands by pressing the **Tab** key twice.

The command modes that are available for system configuration depend on the product packages installed. Base package system management command modes include:

- Root Mode
- Physical Interface Mode
- IP Interface Mode
- OSPF Global Configuration Mode
- Import and Export OSPF Route Filter Modes
- RIP Configuration Mode
- Import and Export RIP Route Filter Modes
- Slot Mode

If the FastFlow Broadband Provisioning Manager is installed on your Cuda 12000 IP Access Switch, additional command modes are available. Refer to the *FastFlow Broadband Provisioning Manager CLI-based Administration Guide* and the *FastFlow Broadband Provisioning Manager CLI Reference Guide* for more information on FastFlow Broadband Provisioning Manager command modes.

Global Commands

Global commands can be used anywhere in the CLI, regardless of your current command mode. Table 1-1 lists global commands as they appear when you type **help** at the command prompt. Note that the **help** command output displays many commands in their abbreviated form.

Table 1-1 Global Commands

Command	Description
basmonitor	Starts the system monitor.
boot	Enables, disables, or reboots a module in an application slot.
clear	Clears a specified system resource (depending on the specified argument), such as ARP cache or statistics counters.
cm-filter-default	Sets the default cable modem filter values for the system.
cpe-control	Sets the default subscriber management values for the system.
connect	Connects you to another Cuda 12000 IP Access Switch chassis.
echo	Echoes a comment so that it displays.
enable	Enables a user session.
help	Displays CLI command help.
interface	Changes you to interface mode.
ip	Configures IP parameters for the system.
no	Specifies the no form of a command.
ping	Enables you to the send an ICMP echo request packet to a destination to determine if it is reachable.
prov-server	Changes you to provisioning server mode. This command is useful only if the FastFlow Broadband Provisioning Manager is installed on your Cuda 12000 IP Access Switch.
q	Shortened form of **quit**.
quit	Enables you to exit from the CLI.
root	Changes you to root mode.
router	Changes you to router mode.

Table 1-1 Global Commands

Command	Description
server	Shortened form of **prov-server**.
set	Sets several user session parameters.
show	Specifies the show form of a command, which provides a read-only view of configuration parameters and other information.
sleep	Delays the display of the CLI prompt for a specified number of seconds.
slot	Changes you to slot mode.
source	Executes a script file.
talk	Enables and disables sending of broadcast messages. This command also allows you to send a message.
traceroute	Traces an IP route from a source to a destination.
up	Moves you up one level in the command mode hierarchy.

Root Mode

Root is the top-level mode in the CLI administration console; all other modes run within this mode. From within root mode you can access second-level command modes, such as slot configuration mode. To enter root mode from within any configuration mode, type **root**.

Table 1-2 lists available root commands as they appear when you type **help** at the command prompt. Global commands are not listed and can be found in Table 1-1 on page 28. Note that the **help** command output displays many commands in their abbreviated form.

Table 1-2 Root Mode Commands

Command	Description
aaa	Configures TACACS+ and RADIUS network authentication.
access-list	Creates an access list, which consists of IP filtering rules.
access-profile	Creates an access profile for a user.
account	Creates user accounts.
alarm-throttle	Configures alarm delivery and threshold parameters.
aux-device	Configures hardware alarm and clocking parameters.
bridge-group	Creates a bridge group.
bridge-timeout	Configures timers for bridge group broadcast flows.
ccdown	Shuts down the control module.
chassis	Configures multi-chassis support parameters.
chassis-config	Configures local chassis parameters.
chassis-fault	Configures chassis alarms.
cm-filter	Creates a cable modem filter.
db-check	Validates provisioning database information.
db-connect	Configures access to the provisioning database.
event-config	Configures event reporting, throttling, and syslog parameters.
event-log	Empties the event log.
http-server	Starts and stops the Web server.
lookup	Controls the Jini lookup service on the chassis.

Table 1-2 Root Mode Commands

Command	Description
modulation-profile	Configures modulation profiles, which contain burst properties for upstream data channels.
privacy	Configures X.509 certificate parameters for BPI plus.
radius-server	Configures a RADIUS authentication server.
reset	Reboots a module.
save	Saves the system configuration for all slots to persistent storage.
snmp-server	Configures the SNMP agent.
tacacs-server	Configures the TACACS+ server.
traffic-relay	Configures traffic relay for processes (such as servers) on the chassis.

Physical Interface Mode

Physical interface mode allows for the administration of a specified interface, including interface-specific configuration and information displays. To enter this mode, you must specify the chassis/slot/port-number *(c/s/i)* combination that identifies the physical interface that you want to configure. After you enter this mode, all configuration that you perform pertains to the interface that you specified.

To enter interface configuration mode, type **interface** <c/s/i> from within any configuration mode. The CLI automatically displays the type of interface for the specific c/s/i. The following tables list interface commands by type:

- Table 1-3 — Lists available DOCSIS interface mode commands as they appear when you type **help** at the command prompt.
- Table 1-4 — Lists available Ethernet interface mode commands as they appear when you type **help** at the command prompt.
- Table 1-5 — Lists available POS interface mode commands as they appear when you type **help** at the command prompt.

Keep in mind that the **help** command output displays many commands in their abbreviated form. Also keep in mind that global commands are not listed in any of these tables and can be found in Table 1-1 on page 28.

NOTE: *The commands displayed via* ***help*****,** ***?*****,** *and through the double Tab action are relevant to the selected interface.*

Table 1-3 DOCSIS Interface Mode Commands

Command	Description
access-class	Applies an access list to the current interface.
access-list	Creates an access list, which consists of IP filtering rules.
admission-control	Enables and disables admission control.
analyzer	Enables the protocol analyzer.
arp	Sets the ARP timeout parameter.
bootp-policy	Defines BOOTP request policies.
cable	Optional prefix to commands in this mode.

Table 1-3 DOCSIS Interface Mode Commands

Command	Description
cm	First element in various cable modem and subscriber management commands, such as **cm modify active**, **cm reset**, and so on.
cm-filter	Creates a cable modem filter.
cm-offline	Configures several offline cable modem parameters for the current interface.
dhcp-authority	Adds a DHCP authority range. The command also enables and disables DHCP authority.
dhcp-policy	Configures several parameters in the DHCP packet and determines the list of servers to which the DHCP packet is sent.
dhcp-relay	Configures various DHCP relay agent options, such as the IP addresses of cable modem, CPE, and MTA gateways.
downstream	Configures the downstream channel.
flap-list	Controls the size of the flap list.
insertion-interval	Configures the modem insertion interval.
link-trap	Enables link traps for the interface.
map-timer	Configures the map timer interval.
modulation-profile	Configures modulation profiles, which contain burst properties for upstream data channels.
periodic-ranging-interval	Configures how the interface periodically invites modems to range.
plant-delay	Configures the estimated plant propagation delay.
pll-state	Configures the phase lock loop state for the interface.
privacy	Configures BPI plus parameters for the interface.
proxy-arp	Enable proxy ARP on the interface.
qos	Enables SNMP and cable modem access to the QoS tables on the CMTS.
ranging-attempts	Configures the number of times that a cable modem is invited to range before being removed from the system.
shared-secret	Configures a shared secret on the current CMTS interface.
shutdown	Enables you to administratively shut down an interface.

Table 1-3 DOCSIS Interface Mode Commands

Command	Description
spectrum-group	Configures spectrum group rules.
sync-interval	Configures the time interval between synchronization message transmissions on the downstream channel.
trace-log	Configures event logging for the interface.
ucd-interval	Configures the time interval between transmission of successive Upstream Channel Descriptor (UCD) messages for each upstream channel.
upstream	Configures upstream channels.

Table 1-4 Ethernet Interface Mode Commands

Command	Description
access-class	Applies an access list to the current interface.
access-list	Creates an access list, which consists of IP filtering rules.
add	Adds a static ARP entry for the current interface.
arp	Sets the ARP timeout parameter.
bootp-policy	Defines BOOTP request policies.
dhcp-authority	Adds a DHCP authority range. The command also enables and disables DHCP authority.
dhcp-policy	Configures several parameters in the DHCP packet and determines the list of servers to which the DHCP packet is sent.
dhcp-relay	Configures DHCP relay agent options.
duplex	Configures the duplex mode for the interface (full duplex, half duplex, or auto).
link-trap	Enables link traps for the interface.
negotiation	Configures an Ethernet port to automatically negotiate duplex and speed.
shutdown	Enables you to administratively shut down an interface.
speed	Configures the speed for an Ethernet port.

Table 1-5 POS Interface Mode Commands

Command	Description
access-class	Applies an access list to the current interface.
access-list	Creates an access list, which consists of IP filtering rules.
arp	Sets the ARP timeout parameter.
bootp-policy	Defines BOOTP request policies.
clock-source	Configures the SONET transmission clock source.
crc	Configures cyclic redundancy checking (CRC).
dhcp-authority	Adds a DHCP authority range. The command also enables and disables DHCP authority.
dhcp-policy	Configures several parameters in the DHCP packet and determines the list of servers to which the DHCP packet is sent.
dhcp-relay	Configures DHCP relay agent options.
link-trap	Enables link traps for the interface.
loop	Configures the current interface for loopback testing.
mtu	Configures the maximum transmission unit (MTU) for the current interface.
pos	Configures POS parameters.
ppp	Configures PPP parameters.
shutdown	Enables you to administratively shut down an interface.

IP Interface Mode

IP interface mode allows for the administration of a specified IP interface, including IP interface-specific configuration and information displays. To enter this mode, you must:

1. Enter physical interface mode for the physical interface associated with the IP interface.
2. Issue the **ip address** command. On the command line, you specify the IP address and network mask combination that identifies the IP interface.

In IP address mode, the following commands are available:

- All commands that are available in the associated physical interface mode (DOCSIS, Ethernet, or POS).
- Commands for configuring RIP and OSPF on the interface (**ip rip** commands and **ip ospf** commands).

OSPF Global Configuration Mode

OSPF commands allow you to configure global OSPF (Open Shortest Path First) parameters. The system supports OSPF version 2 as defined in RFC 1583.

OSPF global configuration mode allows you to enable the protocol on a system-wide basis, and set system-wide OSPF parameters — such as router ID — and default OSPF parameters.

All OSPF areas to which you want this system to belong must be configured within this mode. You then assign areas to OSPF-enabled IP interfaces on an individual basis within IP interface mode. For example, if you want three IP interfaces to belong to three separate areas, you must first define the three areas using the **ospf area** command within this configuration mode. You can then use the **ip ospf** command within the IP interface configuration mode to assign the interface to one of the areas.



You can also set OSPF cost and dead-interval on a per-area basis. Configuration on a per-IP-interface basis overrides the same values that you define in OSPF global configuration mode.

To enter OSPF global configuration mode, type **router ospf** from any mode, or type **ospf** from router mode.

Table 1-6 lists available OSPF global commands as they appear when you type **help** at the command prompt. CLI global commands are not listed and can be found in Table 1-1 on page 28. Note that the **help** command output displays many commands in their abbreviated form.

Table 1-6 OSPF Global Configuration Mode Commands

Command	Description
asbr	Configures the Cuda 12000 IP Access Switch as an Autonomous System Boundary Router (ASBR).
export	Changes you to router export mode.
import	Changes you to router import mode.
ospf	Configures an OSPF area.
ospf-vi	Configures an OSPF virtual interface.
report	Enables sending of OSPF neighbor state and OSPF virtual neighbor state events.
router-id	Configures the OSPF router ID.

Import and Export OSPF Route Filter Modes

Route filters control the flow of routes to and from the routing table. Import route filters control which routes are stored in the routing table. Export filters control which routes are advertised to other routers. You can define route filters to control the flow of both OSPF and RIP routes.

To create OSPF import route filters, enter import mode from within router:ospf mode, or type **router ospf import** from any mode. You can then use the available commands to create route filters to control which OSPF routes the system learns.

To create OSPF export route filters, enter export mode from within router:ospf mode or type **router ospf export** from any mode. You can then use the available commands to create route filters to control which OSPF networks the router advertises to other OSPF routers.

Table 1-7 lists available OSPF import and export route filter commands as they appear when you type **help** at the command prompt. CLI global commands are not listed and can be found in Table 1-1 on page 28. Note that the **help** command output displays many commands in their abbreviated form.

Table 1-7 OSPF Import and Export Route Filter Mode Commands

Command	Description
asbr	Configures the Cuda 12000 IP Access Switch as an Autonomous System Boundary Router.
map-list	Adds a route map to a map list.
match	Specifies criteria that is matched against route entries.
ospf	Configures an OSPF area.
ospf-vi	Configures an OSPF virtual interface.
override	Enables you to override values for import or export filters.
report	Enables sending of OSPF neighbor state and OSPF virtual neighbor state events.
route-map	Creates a route map.
router-id	Configures the OSPF router ID.

RIP Configuration Mode

RIP (Routing Information Protocol) is a broadcast-based protocol used by routers to update routing tables, which include information about the networks that are in their routing tables. The routing table is broadcast to the other routers on the network where RIP is configured over IP.

The Cuda 12000 supports RIP version 2 as defined in RFC 1724. The Cuda can interoperate in a network of both RIPv1 and RIPv2 routers. A network composed of RIPv1 and RIPv2 routers is useful in supporting the transition from older routers to newer routers supporting RIPv2.

In order to exchange RIP routes over an interface you must configure RIP over IP on that interface. After RIP is added to the interface, the Cuda 12000 begins to exchange RIP routes with adjacent RIP routers.

To enter RIP configuration mode, type **router rip** from any mode, or type **rip** from router mode.

RIP configuration mode allows you to enter RIP import and export route filter modes using the **import** and **export** commands. It does not allow you to set global parameters. RIP parameters are configured on a per-IP-interface basis within IP interface mode by means of the **ip rip** command.

Import and Export RIP Route Filter Modes

Route filters control the flow of routes to and from the routing table. Import route filters control which routes are stored in the routing table. Export filters control which routes are advertised to other routers. You can define route filters to control the flow of both OSPF and RIP routes.

To create RIP import route filters, enter import mode from within router:rip mode or type **router rip import** from within any mode. You can then use the available commands to create route filters to control which RIP routes the system learns.

To create RIP export route filters, enter export mode from within router rip mode or type **router rip export** from within any mode. You can then use the available commands to create route filters to control which RIP networks the router advertises to other RIP-enabled routers.

Table 1-8 lists available RIP import and export route filter commands as they appear when you type **help** at the command prompt. CLI global commands are not listed and can be found in Table 1-1 on page 28. Note that the **help** command output displays many commands in their abbreviated form.

Table 1-8 RIP Import and Export Route Filter Mode Commands

Command	Description
map-list	Adds a route map to a map list.
match	Specifies criteria that is matched against route entries.
override	Configures the override values for import or export filters.
route-map	Creates a route map.

Slot Mode

Slot mode provides access to slot-specific commands. To enter this mode, you must specify a chassis/slot (c/s) combination that identifies the slot that you want to administer. Within this mode, you can do the following:

- Persist (save) configuration for the current module, or all modules in the system
- Configure and show trace log activity for the current slot
- Reset the module contained in the slot, or all modules in the chassis.

To enter slot mode, enter **slot** <chassis/slot> from within any mode. Table 1-9 lists available slot mode commands as they appear when you type **help** at the command prompt. CLI global commands are not listed and can be found in Table 1-1 on page 28. Note that the **help** command output displays many commands in their abbreviated form.

Table 1-9 Slot Mode Commands

Command	Description
copy	Downloads a file from a TFTP server to flash.
cpu-utilization	Enables CPU utilization on the module.
filter-aging	Configures IP packet filtering for all interfaces in the slot.
reset	Reboots a module.
save	Saves the system configuration for all slots to persistent storage.
trace-log	Configures event logging for the slot.

2 CUDA 12000 BASE SYSTEM COMMANDS

This section contains an alphabetical listing of CLI commands that comprise the Cuda 12000 base network management system shipped with the system. The following information is provided for each command:

- Description
- Mode
- Syntax
- Arguments
- Example
- Related Commands

aaa authentication

This command enables TACACS+ and RADIUS network access authentication.

Mode

`root`

Syntax

`aaa authentication login default {local | tacacs+ | radius}`

Arguments

{**local | tacacs+ | radius**} Specifies one of the following values:

- local – Authenticate network access locally. Do not use TACACS+ or RADIUS to manage network access authentication.
- tacacs+ – Enables TACACS+ to manage network access authentication.
- radius – Enables RADIUS to manage network access authentication.

Example

This example enables TACACS+ to manage network access security authentication.

```
cli:192.168.208.3:root# aaa authentication login default tacacs+
```

Related Commands

- **show aaa** on page 368
- **tacacs-server** on page 585
- **radius-server** on page 346

access-class

Applies a specified access list to the current interface. Access lists are used to filter, control, and restrict the flow of incoming and outgoing traffic across an interface.

You must first use the **access-list** command to create one or more access-lists comprised the filtering criteria, after which you can then apply those lists to a particular interface using the **access-class** command.

Use the **no access-list** command to delete the association of a specific access list with a particular interface. Use the **show access-list** command to display the access lists that are currently applied to a particular interface.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
access-class <access list number> {in | out} priority <priority>
```

Arguments

<access list number>	Index number of the access list that you want to apply to the current interface.
{**in** | **out**}	Specifies one of the following: ■ in – Applies the specified access list to incoming traffic received on this interface. ■ out – Applies the specified access list to outgoing traffic from this interface.
<priority>	Sets the priority within the access class for the specified access list.

Example

The following example applies access list 1 to cable interface 1/1/1:

```
cli# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli# access-class 1 out priority 1
cli# show access-class

Access List      Priority Row
List   Direction          Status
Number
------ --------- -------- ------
     1       out        1      1

row count: 1

cli# show access-class

Access List      Priority Row
List   Direction          Status
Number
------ --------- -------- ------


     1       out        1      1
```

Related Commands

- **show access-class** on page 369
- **no access-class** on page 260
- **no access-list** on page 262

access-list

Use this command to create a new access list or modify an existing one. Access lists contain filtering rules that define both the match criteria and the actions to take for matching packets. You can create access that control the flow of IP, TCP, and UDP traffic.

Use the **show access-list** command to display access lists and associated filtering rules. Use the **no access-list** command to delete one or all of the access lists that you have previously defined.

Mode

```
root
interface:cable:csi(<c/s/i>)
```

Syntax

Use the following syntax to create an IP access list:

```
access-list <access list number> {permit | deny} <rule number>
  ip {<source IP address> <source IP mask> | host <ip address> |
  host <ip address> | any} {<dest IP address><dest IP mask> | any}
  {<dest IP address> <dest IP mask> | host <dest ip address> | any}
  [tos <tos> <tos mask>] [change-tos <tos>]
```

Use the following syntax to create a TCP access list:

```
access-list <access list number> {permit | deny} <rule number>
  tcp [{<source IP address> <source IP mask> | host <ip address> | any}
  {<dest IP address><dest IP mask> | host <dest ip address> | any}
  {<ip address> <IP mask> | host <ip address> |
  <operator> <port> [<port>] | any}]
  {<IP address> <mask>| any| host <ip address> [ <operator> <port> [<port>]]
  [established] [tos <tos> <tos mask>] [change-tos <tos>]
```

Use the following syntax to create a UDP access list:

```
access-list <access list number> {permit | deny} <rule number>
  udp {<source IP address> <source IP mask> | host <ip address> | any}
  {<dest IP address><dest IP mask> | host <dest ip address> | any}
  {<ip address> <IP mask> | host <ip address> | <operator> <port>
  [<port>] | any}]
  {<IP address> <mask> | any | host <ip address> [<operator><port>
  [<port>]]
  [established] [tos <tos> <tos mask>] [change-tos <tos>]
```

Arguments

<access list number>	Index number that identifies this list. Valid range: 1–65535.
{**permit** | **deny**}	Specifies one of the following values: ■ permit – Permits access if packets match access list criteria. ■ deny – Denies access if packets match access list criteria.
<rule number>	Number identifying the precedence of this access list. Smaller rule numbers result in greater precedence. This means that an access list with a lesser rule number is applied against the interface first.
<source ip address>	IP address seen in the source IP address field of the protocol header. A value of **any** acts as a wildcard.
<source ip mask>	Source IP address network mask, if you specified a specified address.
host	Host address if filtering on a specific IP host.
<dest ip address>	IP address seen in the destination IP address field of the protocol header. A value of **any** acts as a wildcard.
<dest ip address>	Destination IP address network mask, if you specified a specific address.
<operator>	Operand used to compare source and destination ports. You can use the following operands: ■ **lt** (less than) ■ **gt** (greater than) ■ **eq** (equal) ■ **range** (inclusive range)
<port>	TCP or UDP port number. Valid range: 0–65535.

established	For TCP protocol only. Indicates an established TCP connection. Match occurs when the ACK or RST bits of a TCP datagram are set.
<tos>	Type of Service level identified in the IP packet header. Valid Range: 0 – 15.
<tos mask>	Type of Service mask.

Example

The following example uses the **access-list** command to define access list 1 to reject any incoming IP packets then uses the **access-class** command to apply the access list to cable interface 1/1/1.

```
cli# root
mode:root
cli# access-list 1 deny 10 ip any any
cli# show access-list *

List  Ac Rule Prot IP Source       Mask Source     Start Dest  Estab TOS ChTOS
                   IP Dest         Mask Dest       Port  Port  Sync  Msk Log
----- -- ---- ---- --------------- --------------- ----- ----- ----- --- ------
    1 DE   10   ip         0.0.0.0 255.255.255.255             False   0      0
                           0.0.0.0 255.255.255.255             False   0  False
..... .. .... .... ............... ............... ..... ..... ..... ... ......

row count: 1
cli# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli# access-class 1 in priority 1
cli#
```

Related Commands

- **access-class** on page 45
- **no access-list** on page 262
- **show access-list** on page 370

add arp

Adds a static ARP (Address Resolution Protocol) entry to the ARP cache on the current Ethernet interface. Each interface maintains its own ARP cache that can store up to 4,096 entries.

Each ARP entry provides a mapping between a 32-bit IP address and a 48-bit MAC address. The interface uses this mapping to resolve the IP address or MAC address when provided with only one or the other. A static ARP entry permanently associates a MAC address with an IP address until you manually remove the entry from the selected interface using the **del arp** command.

Mode

```
interface:ethernet:csi(<c/s/i>)
```

Syntax

```
add arp <ip-address> <mac-address>
```

Arguments

<ip-address>	IP address that you want to map to the MAC address.
<mac-address>	MAC address that you want to map to the IP address.

Example

This example adds a static entry to the ARP cache for interface 1/11/1. This configures interface 1/11/1 to always resolve MAC address 00:50:41:de:8b:29 to IP address 172.31.1.1.

```
cli:root# interface 1/11/1
mode: interface:ethernet:csi(1/11/1)
cli:interface:ethernet:csi(1/11/1)# add arp 172.31.1.1 00:50:41:de:8b:29
cli:interface:ethernet:csi(1/11/1)# show arp

IP Address       MAC Address        Type
---------------- ------------------ ----------
      172.31.1.1 00:50:41:de:8b:29      static
```

Related Commands

- **del arp** on page 138
- **show arp** on page 374

admission-control

Enables or disables the admission control function of the current CMTS interface. Admission control allocates HFC interface bandwidth to service flows, and prevents admission of flows when bandwidth is unavailable.

Mode

```
interface:cable:csi(<c/s/i>):#
```

Syntax

```
admission-control {enable | disable}
```

Arguments

{**enable** | **disable**}	Specifies one of the following values: ■ enable – Enables admission control. ■ disable – Disables admission control. By default, admission control is disabled.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# admission-control disable
CMTS Admission Control:              Disable

cli:192.168.208.3:interface:cable:csi(1/1/1)# show admission-control
CMTS Admission Control:              Disable
```

Related Commands

- **show admission-control** on page 372
- **upstream <port number> voice-bw-reserve** on page 614

alarm-throttle

Configures the following parameters:

- Alarm delivery interval
- Alarm threshold, which is the maximum number of traps to deliver during an alarm delivery interval

Use the no form of the command to set these parameters to 0.

Mode

```
root
```

Syntax

```
alarm-throttle {alarms <number> | interval <number> | default}

no alarm-throttle
```

Arguments

{**alarms** <number> | **interval** <number> | **default**}	Specifies one of the following values: ■ alarms <number> – The alarm threshold, which is the maximum number of traps that the Cuda 12000 can send during an alarm delivery interval. Values range from 0 to 65535. The default is 300. ■ interval <number> – Specifies the alarm delivery interval in seconds. Values range from 0 to 65535. The default is 3. ■ default – Sets the alarm threshold and alarm delivery interval to their default values.

Example

```
cli:192.168.208.3:root# alarm-throttle alarms 400
cli:192.168.208.3:root# alarm-throttle interval 10
cli:192.168.208.3:root# show alarm-throttle
Alarm Delivery Interval : 10
Max Alarms per Interval : 400
```

Related Commands

- **show alarm-throttle** on page 373

arp timeout

Sets the timeout, in seconds, for dynamic ARP cache entries associated with an interface. When the cache timeout value is exceeded, the Cuda 12000 flushes out-of-date cache entries from the ARP cache. The cache contains IP addresses and their associated MAC addresses that were stored in response to an ARP reply.

Use the no form of the command to set the ARP timeout to the default (600 seconds).

Mode

```
interface:<type>:csi(<c/s/i>):#
```

Syntax

arp timeout <number>

no arp timeout

Arguments

<number>	The ARP timeout value in seconds. Values range from 0 to 604800. The default is 600. Specify 0 if you do not want ARP cache entries to expire under any circumstances.

Example

```
cli:192.168.208.3:interface:cable:csi(1/2/1)# show arp timeout
ARP Aging                              Enabled
ARP Timeout                                600

cli:192.168.208.3:interface:cable:csi(1/2/1)# arp timeout 700
cli:192.168.208.3:interface:cable:csi(1/2/1)# show arp timeout
ARP Aging                              Enabled
ARP Timeout                                700
```

Related Commands

- **show arp timeout** on page 375

asbr

This global Open Shortest Path First command configures the router as an Autonomous System Border Router.

Mode

```
router:ospf
```

Syntax

asbr

Example

```
cli:root# router ospf
mode: router:ospf
cli:router:ospf# asbr
cli:router:ospf# show ospf

Admin Status                          Enabled
TOS Support                             False
Router Id                           201.1.1.1
ABR Status                              False
ASBR Status                              True

Report ospf-nbr-state                 Enabled
Report ospf-virt-nbr-state            Enabled
```

Related Commands

- **no asbr** on page 263
- **router ospf** on page 359
- **show ospf** on page 489

aux-device ac-monitor

Sets the assertion level of the signal that reports AC current faults to the management module.

Mode

`root`

Syntax

`aux-device ac-monitor fault-level {active-high | active-low}`

Arguments

{**active-high** | **active-low**}	Specifies one of the following assertion levels: ■ active-high – The signal reports the assertion state as a logic one state. ■ active-low (the default) – The signal reports the assertion state as a logic zero state. Change to active-high only if the AC unit uses an active high signal to indicate a fault.

Example

```
cli:root# aux-device ac-monitor fault-level active-high
cli:root# show aux-device ac-monitor
Assert Ps AC Fault              active-high
```

Related Commands

- **aux-device backplane-clock-a** on page 56
- **aux-device backplane-clock-b** on page 57
- **aux-device db15 alarm** on page 58
- **aux-device dc-monitor** on page 62
- **aux-device fan-rotation** on page 63
- **aux-device fan-temp** on page 64
- **aux-device ps-temp** on page 65
- **chassis-fault** on page 80
- **show aux-device** on page 376

aux-device backplane-clock-a

Configures clock A on the Cuda 12000 backplane. The backplane has two clocks: A and B. Refer to the description of the **aux-device backplane-clock-b** command for information on configuring clock B.

Mode

`root`

Syntax

`aux-device backplane-clock-a {bits-a | bits-b | internal | none}`

Arguments

{**bits-a | bits-b | internal | none**}	Specifies one of the following clock sources: ■ bits-a – External BITS-A clock acts as clock source. ■ bits-b – External BITS-B clock acts as clock source. ■ internal – Internal Cuda 12000 clock source (Stratum-3 oscillator). ■ none – No clock source. Refer to the *Cuda 12000 IP Access Switch Installation Guide* for information on connecting the switch to external BITS-A and BITS-B clock sources.

Example

```
cli:root# aux-device backplane-clock-a bits-a
```

Related Commands

- **aux-device backplane-clock-b** on page 57
- **show aux-device** on page 376

aux-device backplane-clock-b

Configures clock B on the Cuda 12000 backplane. The backplane has two clocks: A and B. Refer to the description of the **aux-device backplane-clock-a** command for information on configuring clock A.

Mode

`root`

Syntax

`aux-device backplane-clock-b {bits-a | bits-b | internal | none}`

Arguments

{**bits-a | bits-b | internal | none**}	Specifies one of the following clock sources: ■ bits-a – External BITS-A clock acts as the clock source. ■ bits-b – External BITS-B clock acts as the clock source. ■ internal – Internal Cuda 12000 clock source (Stratum-3 oscillator). ■ none – No clock source. Refer to the *Cuda 12000 IP Access Switch Installation Guide* for information on connecting the switch to external BITS-A and BITS-B clock sources.

Example

```
cli:root# aux-device backplane-clock-b internal
```

Related Commands

- **aux-device backplane-clock-a** on page 56
- **show aux-device** on page 376

aux-device db15 alarm

Configures the alarms out port, which is a DB-15 connector on the Cuda 12000 chassis rear panel. You can configure the Cuda 12000 to send specific types of alarm signals out this connector to notify an external device that a fault has occurred. The command disables sending of specific alarms. Use the no form of the command to enable sending of specific alarms.

Mode

root

Syntax

```
aux-device db15 alarm
 {blue | clock [bits-a] [bits-b] [red-alarm] |
  power-alarm [backplane-power] [backplane-pwr-a] [backplane-pwr-b]
  [local-pwr-a] [local-pwr-b] [ps-ac] [ps-dc] | power-fail-A |
  power-fail-B | red [bits-a] [bits-b] [red-alarm] |
  system [backplane] [backplane-power] [backplane-temp]
  [fan-rotation] [fan-temp] [local-pwr-a] [local-pwr-b] [ps-ac]
  [ps-dc] [ps-temp] [red-alarm] | temp [backplane-temp] [fan-temp]
  [processor-temp] [ps-temp] | yellow}

no aux-device db15 alarm
 {blue | clock [bits-a] [bits-b] [red-alarm] |
  power-alarm [backplane-power] [backplane-pwr-a] [backplane-pwr-b]
  [local-pwr-a] [local-pwr-b] [ps-ac] [ps-dc] | power-fail-A |
  power-fail-B | red [bits-a] [bits-b] [red-alarm] |
  system [backplane] [backplane-power] [backplane-temp]
  [fan-rotation] [fan-temp] [local-pwr-a] [local-pwr-b] [ps-ac]
  [ps-dc] [ps-temp] [red-alarm] | temp [backplane-temp] [fan-temp]
  [processor-temp] [ps-temp] | yellow}
```

Arguments

blue	Blue alarms. See the description of the **chassis-fault** command on page 80 for more information on blue alarms.
clock [**bits-a**] [**bits-b**] [**red-alarm**]	Specifies one or more of the following alarms: ■ bits-a – BITS-A clock alarms. ■ bits-b – BITS-B clock alarms. ■ red-alarm – Red alarms.
power-alarm [**backplane-power**] [**backplane-pwr-a**] [**backplane-pwr-b**] [**local-pwr-a**] [**local-pwr-b**] [**ps-ac**] [**ps-dc**]	Specifies one or more of the following alarms: ■ backplane-power – Backplane power alarms. ■ backplane-pwr-a – Backplane power A alarms. ■ backplane-pwr-b – Backplane power B alarms. ■ local-pwr-a – Local power A alarms. ■ local-pwr-b – Local power B alarms. ■ ps-ac – AC power supply alarms. ■ ps-dc – DC power supply alarms.
power-fail-A	Local power failure A alarms.
power-fail-B	Local power failure B alarms.
red [**bits-a**] [**bits-b]** [**red-alarm**]	Specifies one or more of the following alarms: ■ bits-a – BITS-A clock alarms. ■ bits-b – BITS-B clock alarms. ■ red-alarm – Red alarms. See the description of the **chassis-fault** command on page 80 for more information on red alarms.

system [**backplane**] [**backplane-power**] [**backplane-temp**] [**fan-rotation**] [**fan-temp**] [**local-pwr-a**] [**local-pwr-b**] [**ps-ac**] [**ps-dc**] [**ps-temp**] [**red-alarm**]	Specifies one or more of the following alarms: ■ backplane – Backplane system alarms. ■ backplane-power – Backplane power alarms. ■ backplane-temp – Backplane temperature alarms. ■ fan-rotation – Fan rotation alarms. ■ fan-temp – Fan temperature alarms. ■ local-pwr-a – Local power A alarms. ■ local-pwr-b – Local power B alarms. ■ ps-ac – AC power supply alarms. ■ ps-dc – DC power supply alarms. ■ ps-temp – Power supply temperature alarms. ■ red-alarm – Red alarms.
temp [**backplane-temp**] [**fan-temp**] [**processor-temp**] [**ps-temp**]	Specifies one or more of the following alarms: ■ backplane-temp – Backplane temperature alarms. ■ fan-temp – Fan temperature alarms. ■ processor-temp – Processor (CPU) alarms. ■ ps-temp – Power supply temperature alarms.
yellow	Yellow alarms. See the description of the **chassis-fault** command on page 80 for more information on yellow alarms.

Example

```
cli:192.168.208.3:root# no aux-device db15 alarm blue
cli:192.168.208.3:root# show aux-device db15
Temp Alarm                backplane-temp-fault              enabled
                          processor-temp-fault              enabled
                          ps-temp-fault                     enabled
                          fan-temp-fault                    enabled
Sys Alarm                 backplane-system-fault            enabled
                          backplane-temp-fault              enabled
                          backplane-power-fault             enabled
                          local-pwr-a-fault                 enabled
                          local-pwr-b-fault                 enabled
                          red-alarm-fault                   enabled
                          ps-temp-fault                     enabled
                          ps-ac-fault                       enabled
                          ps-dc-fault                       enabled
                          fan-temp-fault                    enabled
                          fan-rotation-fault                enabled
Red Alarm                 bits-a-fault                      enabled
                          bits-b-fault                      enabled
                          red-alarm-fault                   enabled
Blue Alarm                blue-alarm-fault                  enabled
Yellow Alarm              yellow-alarm-fault                enabled
Power Alarm               local-pwr-a-fault                 enabled
                          local-pwr-b-fault                 enabled
                          backplane-power-fault             enabled
--More--
```

Note that "enabled" indicates that the alarm will be sent, and "disabled" indicates that the alarm will not be sent.

Related Commands

- **aux-device ac-monitor** on page 55
- **aux-device backplane-clock-a** on page 56
- **aux-device backplane-clock-b** on page 57
- **aux-device dc-monitor** on page 62
- **aux-device fan-rotation** on page 63
- **aux-device fan-temp** on page 64
- **aux-device ps-temp** on page 65
- **chassis-fault** on page 80
- **show aux-device** on page 376

aux-device dc-monitor

Sets the assertion level of the signal that reports DC current faults to the management module.

Mode

root

Syntax

aux-device dc-monitor fault-level {active-high | active-low}

Arguments

{**active-high** | **active-low**}	Specifies one of the following assertion levels: ■ active-high – The signal reports the assertion state as a logic one state. ■ active-low (the default) – The signal reports the assertion state as a logic zero state. Change to active-high only if the DC unit uses an active high signal to indicate a fault.

Example

```
cli:root# aux-device dc-monitor fault-level active-high
cli:root# show aux-device dc-monitor
Assert Ps DC Fault              active-high
```

Related Commands

- **aux-device ac-monitor** on page 55
- **aux-device backplane-clock-a** on page 56
- **aux-device backplane-clock-b** on page 57
- **aux-device db15 alarm** on page 58
- **aux-device fan-rotation** on page 63
- **aux-device fan-temp** on page 64
- **aux-device ps-temp** on page 65
- **chassis-fault** on page 80
- **show aux-device** on page 376

aux-device fan-rotation

Sets the assertion level of the signal that reports fan rotation faults to the management module.

Mode

`root`

Syntax

`aux-device fan-rotation fault-level {active-high | active-low}`

Arguments

{**active-high** | **active-low**}	Specifies one of the following assertion levels: ■ active-high – The signal reports the assertion state as a logic one state. ■ active-low (the default) – The signal reports the assertion state as a logic zero state. Change to active-high only if the fan uses an active high signal to indicate a fault.

Example

```
cli:root# aux-device fan-rotation fault-level active-high
cli:root# show aux-device fan-rotation
Assert Fan Rotation Fault        active-high
```

Related Commands

- **aux-device ac-monitor** on page 55
- **aux-device backplane-clock-a** on page 56
- **aux-device backplane-clock-b** on page 57
- **aux-device db15 alarm** on page 58
- **aux-device dc-monitor** on page 62
- **aux-device fan-temp** on page 64
- **aux-device ps-temp** on page 65
- **chassis-fault** on page 80
- **show aux-device** on page 376

aux-device fan-temp

Sets the assertion level of the signal that reports fan temperature faults to the management module.

Mode

`root`

Syntax

`aux-device fan-temp fault-level {active-high | active-low}`

Arguments

{**active-high** | **active-low**}	Specifies one of the following assertion levels: ■ active-high – The signal reports the assertion state as a logic one state. ■ active-low (the default) – The signal reports the assertion state as a logic zero state. Change to active-high only if the fan uses an active high signal to indicate a fault.

Example

```
cli:root# aux-device fan-temp fault-level active-high
cli:root# show aux-device fan-temp
Assert Fan Temp Fault            active-high
```

Related Commands

- **aux-device ac-monitor** on page 55
- **aux-device backplane-clock-a** on page 56
- **aux-device backplane-clock-b** on page 57
- **aux-device db15 alarm** on page 58
- **aux-device dc-monitor** on page 62
- **aux-device fan-rotation** on page 63
- **aux-device ps-temp** on page 65
- **chassis-fault** on page 80
- **show aux-device** on page 376

aux-device ps-temp

Sets the assertion level of the signal that reports power supply temperature faults to the management module.

Mode

`root`

Syntax

`aux-device ps-temp fault-level {active-high | active-low}`

Arguments

{**active-high** | **active-low**}	Specifies one of the following assertion levels: ■ active-high – The signal reports the assertion state as a logic one state. ■ active-low (the default) – The signal reports the assertion state as a logic zero state. Change to active-high only if the power supply uses an active high signal to indicate a fault.

Example

```
cli:root# aux-device ps-temp fault-level active-high
cli:root# show aux-device ps-temp
Assert Ps Temp Fault             active-high
```

Related Commands

- **aux-device ac-monitor** on page 55
- **aux-device backplane-clock-a** on page 56
- **aux-device backplane-clock-b** on page 57
- **aux-device db15 alarm** on page 58
- **aux-device dc-monitor** on page 62
- **aux-device fan-rotation** on page 63
- **aux-device fan-temp** on page 64
- **chassis-fault** on page 80
- **show aux-device** on page 376

basmonitor

Starts the basmonitor application within the same telnet window. Use this application to monitor agent-level activity as defined by the **trace-log** command.

Use **trace-log** to define the information that you want to monitor for a specific slot; the specified information is then displayed real-time by basmonitor.

Mode

Any.

Syntax

```
basmonitor
```

Related Commands

- **show trace-log** on page 544
- **trace-log** on page 587

boot

Use this command to disable the card, bring it online, or perform a hard reset from the CLI.

Mode

```
root
slot(<c/s>)
```

Syntax

```
boot {enabled | disabled | reset} slot <c/s>
```

Arguments

{**enabled** | **disabled** | **reset**}	Specifies one of the following values: ■ enabled – Brings the module back online. Issuing this command when the module is disabled boots the module. ■ disabled – Brings the module down. Issuing this command disables the module. ■ reset – Hard boots the module in the current slot. This is the equivalent of pressing the reset button on the module faceplate.
slot <*c/s*>	Applies to root mode only. Specifies the slot that you want to reset.

Example

```
cli:172.16.19.10:slot(1/11)# boot reset
cli:172.16.19.10:slot(1/11)#
```

Related Commands

- **reset** on page 350

bootp-policy

Defines policies to control and restrict the forwarding of BOOTP requests. BOOTP is a protocol that allows diskless workstations to boot off of a network server, called a BOOTP server. You can configure the cable interface to deny (drop) a matching BOOTP request or forward it to a list of BOOTP servers.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
bootp-policy <index> {deny mac <mac-address> [mask <mask>]... |
permit <ip-address>... mac <mac-address> [mask <mask>]... }
[description <string>]
```

Arguments

<index>	Index number to assign to this policy.
deny mac <mac-address> [**mask** <mask>]**...**	Specifies a list of MAC addresses that identify network devices that will be denied the ability to issue BOOTP requests. With each MAC address in the list, you have to option to specify a mask. The mask allows you to specify a range of devices. For example, the following MAC address/mask pair would match all addresses starting with 00:02:09: MAC address: 00:02:09:00:00:00 Mask: FF:FF:FF:00:00:00
permit <ip-address>**... mac** <mac-address> [**mask** <mask>]**...**	Specifies a list of IP addresses and MAC addresses. The IP addresses identify BOOTP servers. The MAC addresses identify network devices that are permitted to forward BOOTP requests to the specified servers.
description <string>	Specifies a descriptive text string for this policy.

Example 1

```
cli:interface:cable:csi(1/1/1)# bootp-policy 2 permit 202.199.1.1 202.199.1.2
202.199.1.3 202.199.1.4 mac 00:90:11:00:00:00 mask ff:ff:ff:00:00:00
```

Example 2

```
cli:interface:cable:csi(1/1/1)# bootp-policy 3 deny mac 4a:01:11:00:00:00
mask ff:ff:ff:00:00:00
```

Related Commands

- **no bootp-policy** on page 264
- **show bootp-policy** on page 379

bridge-group

Creates a network-layer bridge group. A network layer bridge group logically associates multiple physical interfaces into a single logical interface. You use the **ip-address** command to assign an IP address to the logical interface (bridge group). Physical interfaces that comprise the group share the assigned IP address.

Specifying a bridge group as a cable modem or CPE gateway on multiple DOCSIS/EuroDOCSIS modules is particularly useful in spanning a single subnet across multiple (DOCSIS/EuroDOCSIS) modules. In this way, cable modems on different modules can belong to the same network; swapping of the cable modems among participating modules becomes *plug and play*.

When you create a bridge group, you can identify it within the system using a number or a name.

Use the **no bridge-group** command to remove a bridge group from the system.

Mode

```
root
```

Syntax

bridge-group <string>

no bridge-group <string>

Arguments

<string>	Specifies an alphanumeric text string that you want to assign as the name of the bridge group. You can enter up to 256 alphanumeric characters.

Example

The following example creates and then enters configuration mode for Bridge Group 1:

```
cli# root
mode: root
cli# bridge-group 1
cli# show bridge-group

Bridge Group: 1

cli# interface bridge-group 1
mode: interface:bridge-group(1)
```

Related Commands

- **show bridge-group** on page 381

bridge-interface

Use this command within interface configuration mode to add a physical interface to the current bridge group.

To add a physical interface to a bridge group, do the following:

- If you haven't already done so, use the **bridge-group** command to create a bridge group.
- Enter interface configuration mode for the specified bridge group using the **interface bridge-group** command.
- While in interface configuration mode, add a physical interface to the current bridge group using the **bridge-interface** command.

Use the **no bridge-interface** command to remove a physical interface from the current bridge group.

Mode

```
interface:bridge-group(bridge group)
```

Syntax

bridge-interface <c/s/i>

no bridge-interface <c/s/i>

Arguments

<c/s/i>	The chassis/slot/interface combination that identifies the physical interface that you want to add to the bridge-group.

Example

```
cli# bridge-group Bridge_1
cli# interface bridge-group Bridge_1
mode: interface:bridge-group(Bridge_1)
cli# bridge-interface 1/1/1
cli# bridge-interface 1/8/1
cli# bridge-interface 1/11/8
cli# show bridge-group Bridge_1
C/S/I:        129/1/0
Bridge Group: Bridge_1

Chassis    Slot       Interface
---------- ---------- ----------
         1          1          1
         1          8          1
         1         11          8

row count: 3
cli#
```

Related Commands

- **show bridge-group** on page 381

bridge-timeout

Configures aging and reply timers for bridge group broadcast flows.

Mode

To configure timers for all bridge groups on all interfaces:

`root`

To configure timers for a specific bridge group:

`interface:bridge-group(<group>)`

To configure timers for a specific interface:

`interface:cable:csi(<c/s/i>)`

Syntax

bridge-timeout {**aging** <number> | **reply** <number>}

Arguments

aging <number>	Specifies the number of minutes to wait before timing out broadcast flows that are idle or are associated with unreachable destinations. Values range from 10 to 20. The default is 10.
reply <number>	Specifies the number of seconds that the Cuda 12000 waits for a broadcast flow reply to its broadcast flow request before removing the broadcast flow. Values range from 2 to 10. The default is 2. Cuda 12000 switches issue broadcast flow requests to find a destination for an unknown packet. To verify that the destination is still reachable, the Cuda sends this packet to every egress port that belongs to the same bridge group or to one egress port. In response to broadcast flow requests, the module that has the egress port associated with the final destination IP address issues a broadcast flow reply.

Example

```
cli:interface:cable:csi(1/1/1)# bridge-timeout reply 9
cli:interface:cable:csi(1/1/1)# bridge-timeout aging 15
cli:interface:cable:csi(1/1/1)# show bridge-timeout
Reply Timeout                                9
Aging Timeout                               15
```

Related Commands

- **bridge-group** on page 70
- **bridge-interface** on page 72
- **show bridge-timeout** on page 383

ccdown

Shuts down the management module. Use this command before powering off the management module or Cuda 12000 system.

Note that shutdown of the management module starts immediately after entering this command. When the management module has shut down all running processes, the controller's LCD displays the message `Stopped`, after which you can safely power down or restart the system.

Only use this command if you have physical access to the system. After issuing this command, you will need to hit the reset button on the management module to reboot the module.

Mode

`root`

Syntax

`ccdown`

chassis

Allows you to configure and control multi-chassis support.

Mode

```
root
```

Syntax

```
chassis {description <string> | group <group-name> |
  mcs {enable | disable}}
```

Arguments

description <string>	Specifies a description of the group. If the string contains spaces, enclose it in quotes.
group <group-name>	Specifies the name of the multi-chassis group to which this chassis belongs. All chassis that belong to the same group can be managed through a common view, without having to connect to each chassis individually. For example, all chassis in a multi-chassis group can be managed through a single Web page in CudaView. Note that the name may not contain spaces.
mcs {**enable** | **disable**}	Enables multi-chassis support (enable) or disables multi-chassis support (disable).

Example

```
cli:root# chassis group group1
cli:root# chassis mcs disable
cli:root# show chassis local
Multi Chassis Service : disable
Host Name   : Tech1
IP Address  : 192.168.208.3
Group Name  : group1
Version     : 3.0.6 CPM3.0 21 2001_07_17_1036
Description : null
```

Related Commands

- **lookup** on page 250
- **show chassis** on page 385

chassis-config

Configures chassis-wide parameters, including chassis-ID and management module primary or secondary role.

Mode

`root`

Syntax

chassis-config <chassis-number> {**chassisid** <number> | **clusterid** <number> | **manager secondary** | **scope** {**chassis** | **cluster**}}

Arguments

<chassis-number>	Number assigned to the chassis for which you want to configure chassiswide parameters. The system ships with a default chassis number of 101.
{**chassisid** <number> | **clusterid** <number> | **manager secondary** | **scope** {**chassis** | **cluster**}}	Specifies one of the following values: ■ chassisid <number> – Specifies the identification number that you assign to a Cuda 12000 chassis. The Cuda uses a multi-range numbering system. Acceptable chassis ID values are 1 to 128, or the number 255. Default: 1. ■ clusterid <number> – Acceptable values are 0 to 2147483647. Default: 0. The default is recommended. ■ manager secondary – Forces the primary management module into the secondary role, thereby forcing the secondary module into the primary role. Note that if you force the management modules to change roles, your network connection is dropped and you must reconnect to the Cuda. ■ scope – Specifies one of these arguments: chassis – Defines the scope of this management module as chassis-wide. This means that the management module can be used to configure only the current system. cluster – Defines the scope of this management module as cluster. This is the default configuration as the current release supports a single-chassis configuration only. This means that the management module can be used to configure the current system only.

Example 1

In this example, the user displays the chassis configuration and sets the chassis ID:

```
cli:root# show chassis-config
Chassis Number:         101
Chassis Id:             1
Cluster Id:             0
Primary Manager Slot:   13
Secondary Manager Slot: None
Scope:                  Cluster
cli:root# chassis-config 101 chassisid 100
cli:root# slot 1/13
mode: slot(1/13)
cli:root:slot(1/13)# save
Saving slot: 1/13 ..
Save request passed for slot 1/13
cli:root# root
mode: root
cli:root# show chassis config
Chassis Number:         101
Chassis Id:             100
Cluster Id:             0
Primary Manager Slot:   13
Secondary Manager Slot: None
Scope:                  Cluster
```

Example 2

In this example, the user forces the primary management module into the secondary role, thereby forcing the secondary management module into the primary role:

```
cli# show chassis-config
Chassis Number:         101
Chassis Id:             1
Cluster Id:             0
Primary Manager Slot:   13
Secondary Manager Slot: 14
Scope:                  Cluster
cli# chassis-config 101 manager secondary
Connection to 192.168.222.200 refused or closed!
```

Related Commands

- **show chassis-config** on page 386

chassis-fault

Enables Cuda 12000 chassis alarms. The no form of the command disables these alarms.

Mode

root

Syntax

chassis-fault [backplane] [backplane-power] [backplane-power-a] [backplane-power-b] [backplane-temp] [bits-a] [bits-b] [blue] [fan-rotation] [fan-temp] [local-pwr-a] [local-pwr-b] [processor-temp] [ps-ac] [ps-dc] [ps-temp] [red-alarm] [yellow]

no chassis-fault [backplane] [backplane-power] [backplane-power-a] [backplane-power-b] [backplane-temp] [bits-a] [bits-b] [blue] [fan-rotation] [fan-temp] [local-pwr-a] [local-pwr-b] [processor-temp] [ps-ac] [ps-dc] [ps-temp] [red-alarm] [yellow]

Arguments

backplane	Enables the alarm that generates when a module asserts a backplane system fault condition. The no form of the command disables the alarm.
backplane-power	Enables the alarm that generates when one or more modules detect an internal power fault. The no form of the command disables the alarm.
backplane-power-a	Enables the alarm that generates when one or more modules detect a Power_A (48V) fault. The no form of the command disables the alarm.
backplane-power-b	Enables the alarm that generates when one or more modules detect a Power_B (48V) fault. The no form of the command disables the alarm.
backplane-temp	Enables the alarm that generates when one or more modules detect a temperature fault. The no form of the command disables the alarm.
bits-a	Enables the alarm that generates when chassis management software detects loss of the BITS-A clock. The no form of the command disables the alarm.
bits-b	Enables the alarm that generates when chassis management software detects loss of the BITS-B clock. The no form of the command disables the alarm.

blue	Enables blue alarms, which one or more modules generate when they detect a blue alarm condition. A blue alarm occurs when an intermediate device that is currently experiencing a red alarm condition passes alarm information along to the device at the opposite end. The no form of the command disables the alarm.
fan-rotation	Enables the alarm that generates when the fan tray in the chassis detects one or more non-rotating fans. The no form of the command disables the alarm.
fan-temp	Enables the alarm that generates when the fan tray in the chassis detects an inlet temperature greater than 50 degrees celsius. The no form of the command disables the alarm.
local-pwr-a	Enables the alarm that generates when chassis management software detects a loss of Power_A (48V). The no form of the command disables the alarm.
local-pwr-b	Enables the alarm that generates when chassis management software detects a loss of Power_B (48V). The no form of the command disables the alarm.
processor-temp	Enables the alarm that generates when chassis management software detects a processor over-temperature condition.
ps-ac	Enables the alarm that generates when the power supply in the chassis detects loss of one or more AC inputs. The no form of the command disables the alarm.
ps-dc	Enables the alarm that generates when the power supply in the chassis detects a DC power out-of-range fault. The no form of the command disables the alarm.
ps-temp	Enables the alarm that generates when the power supply in the chassis detects an over-temperature condition. The no form of the command disables the alarm.
red-alarm	Enables red alarms, which one or more modules generate when they detect a red alarm condition. A red alarm results from loss-of-signal. The no form of the command disables the alarm.
yellow	Enables yellow alarms, which one or more modules generate when they detect a yellow alarm condition. A yellow alarm indicates that a remote device is transmitting a red alarm. The no form of the command disables the alarm.

Example

```
cli:192.168.244.212:root# chassis-fault backplane-power local-pwr-a
cli:192.168.244.212:root# show chassis-fault status

Chassis Fault Status

Bits A Fault                        disabled
Bits B Fault                        disabled
Backplane System Fault              disabled
Backplane Temp Fault                disabled
Backplane Power Fault                   okay
Backplane Power A Fault             disabled
Backplane Power B Fault             disabled
Red Alarm Fault                     disabled
Blue Alarm Fault                    disabled
Yellow Alarm Fault                  disabled
Processor Temp Fault                disabled
Ps Temp Fault                       disabled
Ps AC Fault                         disabled
Ps DC Fault                         disabled
Fan Temp Fault                      disabled
Fan Rotation Fault                  disabled
Local Pwr A Fault                       okay
Local Pwr B Fault                   disabled
```

Related Commands

- **aux-device ac-monitor** on page 55
- **aux-device backplane-clock-a** on page 56
- **aux-device backplane-clock-b** on page 57
- **aux-device db15 alarm** on page 58
- **aux-device dc-monitor** on page 62
- **aux-device fan-rotation** on page 63
- **aux-device fan-temp** on page 64
- **aux-device ps-temp** on page 65
- **show chassis-fault status** on page 387

clear arp-cache

Deletes all non-static entries in the ARP cache. If you issue this command from root mode, you delete all non-static ARP entries associated with all interfaces. If you issue this command from interface mode, you delete only the non-static ARP entries associated with the current interface.

Mode

```
root
interface:<type>:csi(<c/s/i>)
```

Syntax

```
clear arp-cache
```

Example

```
cli:192.168.208.3:interface:cable:csi(1/2/1)# show arp

row count: 13

IP Address       MAC Address        Type
---------------- ------------------ ----------
10.1.1.100       00:90:83:36:82:f1     dynamic
10.1.1.101       00:90:96:00:39:f9     dynamic
10.1.1.102       00:10:95:04:0a:c3     dynamic
10.1.1.103       00:10:95:04:0a:bd     dynamic
10.1.1.104       00:10:95:04:0a:b7     dynamic
10.1.1.105       00:90:96:00:29:71     dynamic
10.1.1.106       00:90:96:00:29:6d     dynamic
10.1.1.107       00:90:83:32:9f:8c     dynamic

cli:192.168.208.3:interface:cable:csi(1/2/1)# clear arp-cache
```

Related Commands

- **add arp** on page 50
- **del arp** on page 138
- **show arp** on page 374

clear counters

Clears all statistical counters on the current POS interface.

Mode

```
root
interface:pos:csi(<c/s/i>)
```

Syntax

```
clear counters
```

Example

The following example clears the counters on POS interface 1/3/1, then displays interface information (including counters). Note that the display includes a field which indicates when the counters were last cleared.

From within Root mode:

```
cli# root
mode: root
cli# clear counters pos 1/3/1
```

From within Interface mode:

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# clear counters
```

Related Commands

- **interface** on page 181
- **show interface pos** on page 437

clear ip igmp group

Deletes multicast group entries that you create with the **ip igmp join-group** command from the IGMP cache. You can also delete these entries with the **no ip igmp** command. Note that:

- To flush the entire cache on the Cuda 12000, issue the command in root mode without specifying a group address.
- To remove a single multicast group from all interfaces on the Cuda 12000, issue the command in root mode and specify the address of the group.
- To remove all multicast groups from an interface, issue the command in interface mode without specifying a group address.
- To remove a single multicast group from an interface, issue the command in interface mode and specify the address of the group.

Mode

```
root and interface
```

Syntax

```
clear ip igmp group [<group-address>]
```

Arguments

<group-address>	Specifies the Class D IP address (for example, 225.3.2.2) of the multicast group entry to be flushed from the IGMP cache.

Example

In this example, the user flushes the entire IGMP cache on the Cuda 12000:

```
cli:192.168.208.3:root# clear ip igmp group
cli:192.168.208.3:root#
```

Related Commands

- **ip igmp** on page 187

clear service-flow log

Deletes service flow logs. You have the option to delete all the logs or a log for a specific cable modem.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
clear service-flow log {all | <mac address>}
```

Arguments

all	Deletes all service flow logs.
<mac address>	Deletes the service flow log for the cable modem identified by the specified MAC address. Specify the MAC address in xx:xx:xx:xx:xx:xx format.

Example

```
cli:interface:cable:csi(1/1/1)# show modem
row count: 13

MAC Address       IP Address      SID  CID  CPE D:U Power  Timing Modem
                                                   (dbMV)        State
----------------- --------------- ---- ---- --- --- ------ ------ ----------
00:90:83:32:9f:8c 201.1.1.106       43    1   0 1:2      0   1644 Registered
00:10:95:04:0a:bd 201.1.1.111       28    1   0 1:2      0   2724 Registered
00:90:83:36:82:ee 201.1.1.108       44    0   0 1:2      0   1243 RegBpiTek
00:90:83:36:82:f1 201.1.1.107       45    1   0 1:2      0   1233 Registered
00:10:95:04:0a:c4 201.1.1.110       29    1   0 1:2      0   2724 Registered
00:10:95:04:0a:b7 201.1.1.109       32    1   0 1:2      0   2725 Registered
00:90:96:00:29:71 201.1.1.112     1015  176   0 1:2      0   2214 DhcpReqRcv
00:90:96:00:29:6d 201.1.1.104     1014  159   0 1:2      0   2212 DhcpReqRcv
00:90:96:00:39:7f 201.1.1.102     1011  231   0 1:2      0   2215 DhcpReqRcv
00:10:95:01:ef:d8 201.1.1.100       10    1   0 1:2      0   2207 Registered
00:10:95:01:f0:05 201.1.1.103       11    1   0 1:2      0   2207 Registered
00:10:95:04:0a:c3 0.0.0.0         1916   34   0 1:2     -9   2724 Ranging
00:90:96:00:39:f9 201.1.1.101     1013    0   0 1:2      0   2215 DhcpReqRcv

cli:interface:cable:csi(1/1/1)# clear service-flow log 00:90:83:32:9f:8c
```

clock-source

Sets the SONET transmission clock source on the current POS interface. SONET packet transmission is synchronized using a line clock source — a clock source received off the network — or an internally generated clock source.

The POS interface is configured to obtain its transmission clock from the network (line clock source) by default. Use the **no clock-source** command to revert back to this default clock source setting.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
clock-source {line | internal}
```

Arguments

line	Configures the interface to use a clock source it receives off the network to synchronize packet transmission. Also referred to as loop timing, this is the default setting.
internal	Configures the interface to use an internally generated clock source for packet transmission.

Example

The following example configures the POS interface in slot 3 to use the internal clock source for synchronizing SONET transmission.

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# clock-source internal
cli#
```

Related Commands

- **interface** on page 181
- **show interface pos** on page 437

cm-filter

Creates a packet filter for upstream or downstream cable modem or CPE traffic. You can associate multiple filters into a filter group by assigning each filter the same group number.

Use the **no cm-filter** command to remove a filter. If you do not specify a filter number, the entire group is deleted.

The arguments are order-sensitive. Specify the arguments in the order listed.

Mode

Any.

Syntax

```
cm-filter <group-number> <filter-number>
{deny | permit} prot {any | tcp | udp | <number>}
[src <ip-address> <mask>]
[dest <ip-address> <mask>]
[tos <tos-value> <mask>]
[src-port {any | <number>}]
[dest-port {any | <number>}]
[tcp-flag {ack | fin | push | reset | syn | urgent}
   tcp-flag-mask {ack | fin | push | reset | syn | urgent}]

no cm-filter <group-number> [<filter-number>]
```

Arguments

<group-number>	Specifies the ID of the filter group to which the filter belongs. Values range from 0 to 60. You can group filters by group ID. Note that a value of 0 means that no filtering should be performed.
<filter-number>	Specifies the index number for the filter within the group. Values range from 0 to 40.
{**deny** | **permit**}	Specifies one of the following values: ■ deny – Drops packets matching the filter criteria. ■ permit – Allows packets matching the filter criteria to pass.

prot {**any** | **tcp** | **udp** | <number>}

Specifies one of the following protocols for the filter:

- any – Any protocol is filtered.
- tcp – TCP packets are filtered.
- udp – UDP packets are filtered.
- <number> – Packets identified by the protocol number are filtered. You can obtain official protocol numbers from the Internet Assigned Numbers Authority (their Web site is www.iana.org). Specify a number from 0 to 256. Note that specifying 256 is the same as specifying "any."

src <ip-address> <mask>

Optional argument, specifying the source IP address and corresponding mask to match. The default value for the address and mask is 0.0.0.0 (matches any source IP address).

dest <ip-address> <mask>

Optional argument, specifying the destination IP address and corresponding mask to match. The default value for the address and mask is 0.0.0.0 (matches any destination IP address).

tos <tos-value> <mask>

Optional argument, specifying the following values:

- <tos-value> – Two-digit hexadecimal number indicating the Type of Service (ToS) value (for example, 0a) to be matched against the ToS value in IP packets. The default is 00.
- <mask> – Two-digit hexadecimal number that specifies the mask to be applied to the ToS value matched in the IP packet (for example 1b). The mask determines which bits are matched (a 0 specifies a match while a 1 specifies no match). The default is 00, which means that the ToS value you specify is matched against all ToS values in IP packets.

src-port {**any** | <number>}

Optional argument, specifying the source TCP or UDP port number to match. Specify one of the following values:

- any – Match any source port.
- <number> – Match a source port number. The range is 0 to 65536 where a value of 65536 (the default) matches any value in the TCP or UDP source port field. Specifying the 65536 value is the same as specifying the **any** argument.

Note that this argument applies to TCP and UDP filters only.

dest-port {**any** | <number>}

Optional argument, specifying the destination TCP or UDP port number to match. Specify one of the following values:

- any – Match any destination port.
- <number> – Match a destination port number. The range is 0 to 65536 where a value of 65536 (the default) matches any value in the TCP or UDP source port field. Specifying the 65536 value is the same as specifying the **any** argument.

Note that this argument applies to TCP and UDP filters only.

tcp-flag {**urgent** | **ack** | **push** | **reset** | **syn** | **fin**}

Optional argument, specifying the value of the TCP flags, which must always be a subset of the TCP flag mask. There is no default (no flags). The options are:

- urgent – The TCP segment is marked urgent.
- ack – The acknowledgement number field in the TCP field segment is significant.
- push – The TCP software must push all the data sent so far through the connection to the receiving application.
- reset – The connection is reset.
- syn – The sequence numbers are resynchronized, marking the beginning of a connection.
- fin – The transmitting CPE device has no more data to transmit.

The TCP flags must always be a subset of the TCP flag mask in order for the packet header to be matched. For example to match all packets where the urgent bit is set, but that are not either syn or fin, the flag values would be "urgent" and the mask would be:

urgent, syn, fin

tcp-flag-mask {**urgent** | **ack** | **push** | **reset** | **syn** | **fin**}

Optional argument, specifying the flag of interest in the TCP header for the packet to match. There is no default (no mask). The options are:

- urgent
- ack
- push
- reset
- syn
- fin

See the description of the **tcp-flag** argument for more information on these flags.

Example

```
cli:root# cm-filter 2 1 deny prot tcp dest 144.133.1.1 255.255.255.0
cli:root# show cm-filter 2 1
Group                                    2
Index                                    1
Src Address               0.0.0.0
Src Mask                  0.0.0.0
Dest Address              144.133.1.0
Dest Mask                 255.255.255.0
Protocol                                tcp
TOS                       00
TOS Mask                  00
Action                                 deny
Matches                                   0

Source Port                            65536
Destination Port                       65536
TCP Flag Values
TCP Flag Mask
```

Related Commands

- **show cm-filter** on page 389

cm-filter-default cm downstream

Specifies the default downstream filter group for cable modems on all CMTS interfaces. You can use the **cm modify cm-downstream** command to override this default setting on a modem-by-modem basis.

Mode

Any.

Syntax

cm-filter-default cm downstream <group-id>

Arguments

<group-id>	Specifies the ID of a filter group, which you create using the **cm-filter** command. The default value is 0, which specifies that no default filter group is available for downstream filtering. Use the **show cm-filter** command to display available filter groups.

Example

```
cli:root# show cm-filter

row count: 1

Group Index Src Address      Src Mask         Dest Address     Dest Mask
----- ----- --------------  ---------------  --------------   --------------
    1     1 101.100.3.0     255.255.255.0    102.100.3.0      255.255.255.0

cli:root# cm-filter-default cm downstream 1
cli:root# show cm-filter-default
CPE DS Filter Group                       0
CPE US Filter Group                       0
CM DS Filter Group                        1
CM US Filter Group                        0
```

Related Commands

- **cm-filter** on page 88
- **cm-filter-default cm upstream** on page 94
- **cm-filter-default cpe downstream** on page 96
- **cm-filter-default cpe upstream** on page 98
- **cm modify cm-downstream** on page 111
- **show cm-filter** on page 389
- **show cm-filter-default** on page 390
- **show interface cable <c/s/i> cm-filter-default** on page 415

cm-filter-default cm upstream

Specifies the default upstream filter group for cable modems on all CMTS interfaces. You can use the **cm modify cm-upstream** command to override this default setting on a modem-by-modem basis.

Mode

Any.

Syntax

cm-filter-default cm upstream <group-id>

Arguments

<group-id>	Specifies the ID of a filter group, which you create using the **cm-filter** command. The default value is 0, which specifies that no default filter group is available for upstream filtering. Use the **show cm-filter** command to display available filter groups.

Example

```
cli:root# show cm-filter

row count: 1

Group Index Src Address      Src Mask         Dest Address     Dest Mask
----- ----- --------------  ---------------  --------------   --------------
    1     1 101.100.3.0     255.255.255.0    102.100.3.0      255.255.255.0

cli:root# cm-filter-default cm upstream 1
cli:root# show cm-filter-default
CPE DS Filter Group                       0
CPE US Filter Group                       0
CM DS Filter Group                        0
CM US Filter Group                        1
```

Related Commands

- **cm-filter** on page 88
- **cm-filter-default cm downstream** on page 92
- **cm-filter-default cpe downstream** on page 96
- **cm-filter-default cpe upstream** on page 98
- **cm modify cm-upstream** on page 113
- **show cm-filter** on page 389
- **show cm-filter-default** on page 390
- **show interface cable <c/s/i> cm-filter-default** on page 415

cm-filter-default cpe downstream

Specifies the default downstream filter group for CPE devices on all CMTS interfaces. You can use the **cm modify cpe-downstream** command to override this default setting on a modem-by-modem basis.

Mode

Any.

Syntax

cm-filter-default cpe downstream <group-id>

Arguments

<group-id>	Specifies the ID of a filter group, which you create using the **cm-filter** command. The default value is 0, which specifies that no default filter group is available for downstream filtering. Use the **show cm-filter** command to display available filter groups.

Example

```
cli:root# show cm-filter

row count: 1

Group Index Src Address      Src Mask         Dest Address     Dest Mask
----- ----- --------------  ---------------  --------------   --------------
    1     1 101.100.3.0      255.255.255.0    102.100.3.0      255.255.255.0

cli:root# cm-filter-default cpe downstream 1
cli:root# show cm-filter-default
CPE DS Filter Group                       1
CPE US Filter Group                       0
CM DS Filter Group                        0
CM US Filter Group                        0
```

Related Commands

- **cm-filter** on page 88
- **cm-filter-default cm downstream** on page 92
- **cm-filter-default cm upstream** on page 94
- **cm-filter-default cpe upstream** on page 98
- **cm modify cm-upstream** on page 113
- **show cm-filter** on page 389
- **show cm-filter-default** on page 390
- **show interface cable <c/s/i> cm-filter-default** on page 415

cm-filter-default cpe upstream

Specifies the default upstream filter group for CPE devices on all CMTS interfaces.
You can use the **cm modify cpe-upstream** command to override this default setting on a modem-by-modem basis.

Mode

Any.

Syntax

cm-filter-default cpe upstream <group-id>

Arguments

<group-id>	Specifies the ID of a filter group, which you create using the **cm-filter** command. The default value is 0, which specifies that no default filter group is available for upstream filtering. Use the **show cm-filter** command to display available filter groups.

Example

```
cli:root# show cm-filter

row count: 1

Group Index Src Address      Src Mask         Dest Address     Dest Mask
----- ----- ---------------- ---------------- --------------   --------------
    1     1 101.100.3.0      255.255.255.0    102.100.3.0      255.255.255.0

cli:root# cm-filter-default cpe upstream 1
cli:root# show cm-filter-default
CPE DS Filter Group                       0
CPE US Filter Group                       1
CM DS Filter Group                        0
CM US Filter Group                        0
```

Related Commands

- **cm-filter** on page 88
- **cm-filter-default cm downstream** on page 92
- **cm-filter-default cm upstream** on page 94
- **cm-filter-default cpe downstream** on page 96
- **cm modify cm-upstream** on page 113
- **show cm-filter** on page 389
- **show cm-filter-default** on page 390
- **show interface cable <c/s/i> cm-filter-default** on page 415

cm-offline clear

Flushes all offline cable modems on the current interface from the CMTS modem tables.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

cm-offline clear

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem
row count: 13
MAC Address       IP Address      SID  CID  CPE D:U Power  Timing Modem
                                                   (dbMV)        State
----------------- --------------- ---- ---- --- --- ------ ------ ----------
00:10:95:04:0a:c3 0.0.0.0           2    0   0 1:2     -9   2725 Ranging
00:90:96:00:29:6d 201.1.1.104       3    0   0 1:2      0   2215 RangComple
00:10:95:04:0a:c4 201.1.1.110       4    1   0 1:2      0   2723 Registered
00:10:95:04:0a:bd 201.1.1.111       5    1   0 1:2      0   2724 Registered
00:90:96:00:29:71 201.1.1.112       6   22   0 1:2      0   2215 DhcpReqRcv
00:10:95:04:0a:b7 201.1.1.109       7    1   0 1:2      0   2725 Registered
00:90:83:32:9f:8c 201.1.1.106       8    1   0 1:2      0   1637 Offline
00:90:96:00:39:7f 201.1.1.102       9    0   0 1:2      0   2215 DhcpReqRcv
00:90:83:36:82:f1 201.1.1.107      10    1   0 1:2      0   1230 Registered
00:90:83:36:82:ee 201.1.1.108      11    0   0 1:2      0   1240 RegBpiTek
00:10:95:01:ef:d8 201.1.1.100      12    1   0 1:2      0   2208 Registered
00:10:95:01:f0:05 201.1.1.103      13    1   0 1:2      0   2209 Registered
00:90:83:32:4a:09 201.1.1.113      14    1   0 1:2      0   1663 Registered

cli:192.168.208.3:interface:cable:csi(1/1/1)# cm-offline clear
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem

row count: 12

MAC Address       IP Address      SID  CID  CPE D:U Power  Timing Modem
                                                   (dbMV)        State
----------------- --------------- ---- ---- --- --- ------ ------ ----------
00:10:95:04:0a:c3 0.0.0.0           2    0   0 1:2     -8   2726 Ranging
00:90:96:00:29:6d 201.1.1.104       3    0   0 1:2      0   2215 DhcpReqRcv
00:10:95:04:0a:c4 201.1.1.110       4    1   0 1:2      0   2722 Registered
00:10:95:04:0a:bd 201.1.1.111       5    1   0 1:2      0   2725 Registered
00:90:96:00:29:71 201.1.1.112       6   22   0 1:2      0   2218 DhcpReqRcv
00:10:95:04:0a:b7 201.1.1.109       7    1   0 1:2      0   2726 Registered
00:90:96:00:39:7f 201.1.1.102       9    0   0 1:2      0   2213 DhcpReqRcv
00:90:83:36:82:f1 201.1.1.107      10    1   0 1:2      0   1231 Registered
00:90:83:36:82:ee 201.1.1.108      11    0   0 1:2      0   1238 RegBpiTek
00:10:95:01:ef:d8 201.1.1.100      12    1   0 1:2      0   2207 Registered
00:10:95:01:f0:05 201.1.1.103      13    1   0 1:2      0   2209 Registered
00:90:83:32:4a:09 201.1.1.113      14    1   0 1:2      0   1663 Registered
```

Related Commands

- **cm-offline persist** on page 102
- **cm-offline timer** on page 103

cm-offline persist

Enables the CMTS to maintain statistics for a cable modem after the modem goes offline. When the cable modem comes back online, the CMTS factors the original statistics (that is, the modem's statistics before it went offline) into the current statistics counters for the cable modem.

By default, the CMTS maintains statistics for cable modems when they go offline.

Use the no form of the command to disable the ability of the CMTS to maintain statistics for cable modems that go offline.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

cm-offline persist

no cm-offline persist

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# cm-offline persist
cli:192.168.208.3:interface:cable:csi(1/1/1)# show cm-offline
Cable Modem Offline Timer                30
Cable Modem Stats Persist              enabled
```

Related Commands

- **cm-offline clear** on page 100
- **cm-offline timer** on page 103
- **show cm-offline** on page 391

cm-offline timer

Specifies the number of days that the CMTS tracks offline cable modems on the current interface. Use the no form of the command to disable the ability of the CMTS to track offline cable modems.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

cm-offline timer <number>

no cm-offline timer

Arguments

<number>	Specifies the number of days that the CMTS tracks offline cable modems. Values range from 0 to 365. The default is 30. If you specify a value of 0, the CMTS will track offline cable modems for an infinite period of time.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# cm-offline timer 35
cli:192.168.208.3:interface:cable:csi(1/1/1)# show cm-offline
Cable Modem Offline Timer                  35
Cable Modem Stats Persist              enabled
```

Related Commands

- **cm-offline clear** on page 100
- **cm-offline persist** on page 102
- **show cm-offline** on page 391

cm cpe-reset

Clears the CPE IP addresses that the CMTS has learned for the specified cable modem.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

To clear CPE IP addresses for a single modem:

cm cpe-reset {<ip-address> | <mac-address> | <sid>}

To clear CPE IP addresses for multiple modems with common hexadecimal MAC address values:

cm cpe-reset <hex-values> **match**

To clear CPE IP addresses for all modems attached to a selected network:

cm cpe-reset <address-string> **match**

Arguments

{<ip-address> | <mac-address> | <sid>}	Specifies one of the following values: ■ <ip-address> – The IP address of the cable modem for which you want to clear IP addresses. ■ <mac-address> – MAC address (in xx:xx:xx:xx:xx:xx format) of the cable modem for which you want to clear IP addresses. ■ <sid> – The service identifier of the cable modem for which you want to clear IP addresses. Use the **show modem** command to determine the IP address, MAC address, and service identifier of the cable modem.
<hex-values>	Specifies the hex values that you want to match, which you enter with "ff" values specified as wildcards. For example, if you want to clear CPE IP addresses for all cable modems from vendor 00:50:72, enter the following command: **cm cpe-reset 00:50:72:ff:ff:ff match**

<address-string>	Specifies the IP address that you want to match, which you enter with a "255" wildcard mask. For example, if you want to clear CPE IP addresses for all cable modems attached to subnet 189.23.3.x, enter the following command: **cm cpe-reset 189.23.3.255 match**
match	Keyword used to define an address or MAC address match.

Example 1

The following example clears CPE IP addresses for a single modem with the SID of 667.

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem

row count: 12

MAC Address       IP Address      SID  CID   CPE   D:U Power  Timing Modem
                                                       (dbMV)        State
----------------- --------------- ---- ----  ---   --- ------ ------ -------
00:90:96:00:29:6d 201.1.1.103      667    1   10   1:2      0   2219 Registered
00:10:95:01:ef:d8 201.1.1.100      673    1   10   1:2      0   2212 Registered
00:10:95:04:0a:c4 201.1.1.102      676    1   10   1:2      0   2724 Registered
00:90:96:00:39:f9 201.1.1.101      668    1   10   1:2      0   2220 Registered
00:10:95:04:0a:b7 201.1.1.105      675    1   10   1:2      0   2729 Registered
00:90:96:00:29:71 201.1.1.104      670    1   10   1:2      0   2217 Registered
00:10:95:04:0a:bd 201.1.1.106      677    1   10   1:2      1   2728 Registered
00:10:95:01:f0:05 201.1.1.107      674    1   10   1:2      0   2211 Registered
00:90:83:36:82:f1 201.1.1.110      669    1   10   1:2      0   1244 Registered
00:90:83:32:9f:8c 201.1.1.111      666    1   10   1:2      0   1653 Registered
00:10:95:04:0a:c3 0.0.0.0           62    0    0   1:2     -8   2729 Ranging

cli:192.168.208.3:interface:cable:csi(1/1/1)# cm cpe-reset 667
```

Example 2

The following example displays the modems attached to cable interface 1/1/1 then uses the match argument to clear CPE IP addresses for all modems with the vendor ID: 00:90:96.

```
cli:interface:cable:csi(1/1/1)# show modem

row count: 11

MAC Address       IP Address      SID  CID   CPE   D:U Power  Timing Modem
                                                      (dbMV)        State
----------------- --------------- ---- ----  ---    --- ------ ------ -------
00:90:96:00:29:6d 201.1.1.103      667    1   10   1:2      0   2219 Registered
00:10:95:01:ef:d8 201.1.1.100      673    1   10   1:2      0   2212 Registered
00:10:95:04:0a:c4 201.1.1.102      676    1   10   1:2      0   2724 Registered
00:90:96:00:39:f9 201.1.1.101      668    1   10   1:2      0   2220 Registered
00:10:95:04:0a:b7 201.1.1.105      675    1   10   1:2      0   2729 Registered
00:90:96:00:29:71 201.1.1.104      670    1   10   1:2      0   2217 Registered
00:10:95:04:0a:bd 201.1.1.106      677    1   10   1:2      1   2728 Registered
00:10:95:01:f0:05 201.1.1.107      674    1   10   1:2      0   2211 Registered
00:90:83:36:82:f1 201.1.1.110      669    1   10   1:2      0   1244 Registered
00:90:83:32:9f:8c 201.1.1.111      666    1   10   1:2      0   1653 Registered
00:10:95:04:0a:c3 0.0.0.0           62    0    0   1:2     -8   2729 Ranging

cli:interface:cable:csi(1/1/1)# cm cpe-reset 00:90:96:ff:ff:ff match
```

Example 3

The following example uses the match argument against the IP address parameter to clear CPE IP addresses for all cable modems on the 201.1.1.0 subnet.

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem

row count: 9

MAC Address       IP Address      SID  CID   CPE   D:U Power  Timing Modem
                                                      (dbMV)        State
----------------- --------------- ---- ----  ---   --- ------ ------ -------
00:90:96:00:29:6d 201.1.1.103     667    1   10   1:2      0   2219 Registered
00:10:95:01:ef:d8 201.1.1.100     673    1   10   1:2      0   2212 Registered
00:10:95:04:0a:c4 201.1.1.102     676    1   10   1:2      0   2724 Registered
00:90:96:00:39:f9 201.1.1.101     668    1   10   1:2      0   2220 Registered
00:10:95:04:0a:b7 201.1.1.105     675    1   10   1:2      0   2729 Registered
00:90:96:00:29:71 201.1.1.104     670    1   10   1:2      0   2217 Registered
00:10:95:04:0a:bd 201.1.1.106     677    1   10   1:2      1   2728 Registered
00:10:95:01:f0:05 201.1.1.107     674    1   10   1:2      0   2211 Registered
00:90:83:36:82:f1 201.1.1.110     669    1   10   1:2      0   1244 Registered
00:90:83:32:9f:8c 201.1.1.111     666    1   10   1:2      0   1653 Registered
00:10:95:04:0a:c3 0.0.0.0           62    0    0   1:2     -8   2729 Ranging

cli:192.168.208.3:interface:cable:csi(1/1/1)# cm cpe-reset 201.1.1.255 match
```

Example 4

The following example uses the match argument to wildcard the complete MAC address and clear CPE IP addresses for all modems.

```
cli:interface:cable:csi(1/1/1)# show modem

row count: 11

MAC Address       IP Address      SID  CID   CPE   D:U Power  Timing Modem
                                                       (dbMV)        State
----------------- --------------- ---- ----  ---    --- ------ ------ -------
00:90:96:00:29:6d 201.1.1.103      667    1   10   1:2      0   2219 Registered
00:10:95:01:ef:d8 201.1.1.100      673    1   10   1:2      0   2212 Registered
00:10:95:04:0a:c4 201.1.1.102      676    1   10   1:2      0   2724 Registered
00:90:96:00:39:f9 201.1.1.101      668    1   10   1:2      0   2220 Registered
00:10:95:04:0a:b7 201.1.1.105      675    1   10   1:2      0   2729 Registered
00:90:96:00:29:71 201.1.1.104      670    1   10   1:2      0   2217 Registered
00:10:95:04:0a:bd 201.1.1.106      677    1   10   1:2      1   2728 Registered
00:10:95:01:f0:05 201.1.1.107      674    1   10   1:2      0   2211 Registered
00:90:83:36:82:f1 201.1.1.110      669    1   10   1:2      0   1244 Registered
00:90:83:32:9f:8c 201.1.1.111      666    1   10   1:2      0   1653 Registered
00:10:95:04:0a:c3 0.0.0.0           62    0    0   1:2     -8   2729 Ranging

cli:interface:cable:csi(1/1/1)# cm cpe-reset ff:ff:ff:ff:ff:ff match
```

Related Commands

- **cm modify learnable** on page 119
- **cpe-control learnable** on page 131
- **show modem** on page 459

cm modify active

Enables the ability of the CMTS to perform subscriber management (IP address limits, filtering, etc.) of CPE devices associated with the specified cable modem. This command overrides the default that the **cpe-control active** command specifies.

Use the no form of the command to disable the ability of the CMTS to perform subscriber management.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
cm modify active {<ip-address> | <mac-address> | <sid>}

no cm modify active {<ip-address> | <mac-address> | <sid>}
```

Arguments

{<ip-address> **|** <mac-address> **|** <sid>}	Specifies one of the following values: ■ <ip-address> – The IP address of the cable modem for which you want to enable or disable subscriber management. ■ <mac-address> – The MAC address (in xx:xx:xx:xx:xx:xx format) of the cable modem for which you want to enable or disable subscriber management. ■ <sid> – The service identifier of the cable modem for which you want to enable or disable subscriber management. Use the **show modem** command to determine the IP address, MAC address, and service identifier.

Example

```
cli:interface:cable:csi(1/1/1)# show modem

row count: 12

MAC Address       IP Address      SID  CID   CPE   D:U Power  Timing Modem
                                                      (dbMV)        State
----------------- --------------- ---- ----  ---   --- ------ ------ -------
00:90:96:00:29:6d 201.1.1.103      67    1    0   1:2      0   2219 Registered
00:10:95:01:ef:d8 201.1.1.100      71    1    0   1:2      0   2209 Registered
00:10:95:04:0a:c4 201.1.1.102      63    1    0   1:2      0   2730 Registered
00:90:96:00:39:f9 201.1.1.101      68    1    0   1:2      0   2219 Registered
00:10:95:04:0a:b7 201.1.1.105      64    1    0   1:2      0   2727 Registered
00:90:96:00:29:71 201.1.1.104      66    1    0   1:2      0   2222 Registered
00:10:95:04:0a:bd 201.1.1.106      61    1    0   1:2      0   2729 Registered
00:10:95:01:f0:05 201.1.1.107      75    1    0   1:2      0   2213 Registered
00:90:96:00:39:7f 201.1.1.108     962    2    0   1:2      0   2221 TftpReqRcv
00:90:83:36:82:f1 201.1.1.110      73    1    0   1:2      0   1235 Registered
00:90:83:32:9f:8c 201.1.1.111      70    1    0   1:2      0   1657 Registered
00:10:95:04:0a:c3 0.0.0.0           62    0    0   1:2     -9   2730 Ranging

cli:interface:cable:csi(1/1/1)# cm modify active 67
```

Related Commands

- **cm modify cm-downstream** on page 111
- **cm modify cm-upstream** on page 113
- **cm modify cpe-downstream** on page 115
- **cm modify cpe-upstream** on page 117
- **cm modify learnable** on page 119
- **cm modify max-ip** on page 121
- **cm modify upstream** on page 123
- **cpe-control active** on page 130
- **show modem** on page 459

cm modify cm-downstream

Assigns a downstream filter group to a cable modem. The value that this command specifies overrides the default that the **cm-filter-default cm downstream** command specifies.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
cm modify cm-downstream <group-id>
{<ip-address> | <mac-address> | <sid>}
```

Arguments

<group-id>	Specifies the ID of a filter group, which you create using the **cm-filter** command. Use the **show cm-filter** command to display available filter groups. Specifying a value of 0 disables downstream filtering for the cable modem.
{<ip-address> | <mac-address> | <sid>}	Specifies one of the following values: ■ <ip-address> – The IP address of the cable modem to which you assign the downstream filter group. ■ <mac-address> – The MAC address (in xx:xx:xx:xx:xx:xx format) of the cable modem to which you assign the downstream filter group. ■ <sid> – The service identifier of the cable modem to which you assign the downstream filter group. Use the **show modem** command to determine the IP address, MAC address, and service identifier.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show cm-filter

row count: 1

Group Index Src Address      Src Mask         Dest Address     Dest Mask
----- ----- ---------------- ---------------- ---------------- --------------
    1     1 0.0.0.0          0.0.0.0          144.133.1.0      255.255.255.0

cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem

row count: 11

MAC Address       IP Address      SID  CID   CPE   D:U Power  Timing Modem
                                                      (dbMV)        State
----------------- --------------- ---- ----  ---   --- ------ ------ -------
00:90:96:00:29:6d 201.1.1.103       67    1    0   1:2      0   2220 Registered
00:10:95:01:ef:d8 201.1.1.100       71    1    0   1:2      0   2209 Registered
00:10:95:04:0a:c4 201.1.1.102       63    1    0   1:2      0   2729 Registered
00:90:96:00:39:f9 201.1.1.101       68    1    0   1:2      0   2220 Registered
00:10:95:04:0a:b7 201.1.1.105       64    1    0   1:2      0   2728 Registered
00:90:96:00:29:71 201.1.1.104       66    1    0   1:2      0   2222 Registered
00:10:95:04:0a:c3 0.0.0.0           62    0    0   1:2     -8   2729 Ranging

cli:192.168.208.3:interface:cable:csi(1/1/1)# cm modify cm-downstream 1 71
```

Related Commands

- **cm-filter** on page 88
- **cm-filter-default cm downstream** on page 92
- **cm modify active** on page 109
- **cm modify cm-upstream** on page 113
- **cm modify cpe-downstream** on page 115
- **cm modify cpe-upstream** on page 117
- **cm modify learnable** on page 119
- **cm modify max-ip** on page 121
- **cm modify upstream** on page 123
- **cpe-control active** on page 130
- **show cm-filter** on page 389
- **show cm-filter-default** on page 390
- **show modem** on page 459

cm modify cm-upstream

Assigns an upstream filter group to a cable modem. The value that this command specifies overrides the default that the **cm-filter-default cm upstream** command specifies.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
cm modify cm-upstream <group-id>
{<ip-address> | <mac-address> | <sid>}
```

Arguments

<group-id>	Specifies the ID of a filter group, which you create using the **cm-filter** command. Use the **show cm-filter** command to display available filter groups. Specifying a value of 0 disables upstream filtering for the cable modem.
{<ip-address> | <mac-address> | <sid>}	Specifies one of the following values: ■ <ip-address> – The IP address of the cable modem to which you assign the upstream filter group. ■ <mac-address> – The MAC address (in xx:xx:xx:xx:xx:xx format) of the cable modem to which you assign the upstream filter group. ■ <sid> – The service identifier of the cable modem to which you assign the upstream filter group. Use the **show modem** command to determine the IP address, MAC address, and service identifier.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show cm-filter
row count: 1
Group Index Src Address      Src Mask         Dest Address     Dest Mask
----- ----- --------------- --------------- --------------- --------------
    1     1 0.0.0.0         0.0.0.0         144.133.1.0      255.255.255.0

cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem

row count: 11

MAC Address       IP Address      SID  CID   CPE   D:U Power  Timing Modem
                                                      (dbMV)        State
----------------- --------------- ---- ----  ---   --- ------ ------ -------
00:90:96:00:29:6d 201.1.1.103      67    1    0   1:2      0   2220 Registered
00:10:95:01:ef:d8 201.1.1.100      71    1    0   1:2      0   2209 Registered
00:10:95:04:0a:c4 201.1.1.102      63    1    0   1:2      0   2729 Registered
00:90:96:00:39:f9 201.1.1.101      68    1    0   1:2      0   2220 Registered
00:10:95:04:0a:b7 201.1.1.105      64    1    0   1:2      0   2728 Registered
00:90:96:00:29:71 201.1.1.104      66    1    0   1:2      0   2222 Registered
00:10:95:04:0a:bd 201.1.1.106      61    1    0   1:2      0   2729 Registered
00:10:95:04:0a:c3 0.0.0.0           62    0    0   1:2     -8   2729 Ranging

cli:192.168.208.3:interface:cable:csi(1/1/1)# cm modify cm-upstream 1 71
```

Related Commands

- **cm-filter** on page 88
- **cm-filter-default cm upstream** on page 94
- **cm modify active** on page 109
- **cm modify cm-downstream** on page 111
- **cm modify cpe-downstream** on page 115
- **cm modify cpe-upstream** on page 117
- **cm modify learnable** on page 119
- **cm modify max-ip** on page 121
- **cm modify upstream** on page 123
- **cpe-control active** on page 130
- **show cm-filter** on page 389
- **show cm-filter-default** on page 390
- **show modem** on page 459

cm modify cpe-downstream

Assigns a downstream filter group to CPE devices that access the network through the specified cable modem. The value that this command specifies overrides the default that the **cm-filter-default cpe downstream** command specifies.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
cm modify cpe-downstream <group-id>
{<ip-address> | <mac-address> | <sid>}
```

Arguments

<group-id>	Specifies the ID of a filter group, which you create using the **cm-filter** command. Use the **show cm-filter** command to display available filter groups. Specifying a value of 0 disables downstream filtering for the CPE devices.
{<ip-address> | <mac-address> | <sid>}	Specifies one of the following values: ■ <ip-address> – The IP address of the cable modem that provides network access to the CPE devices to which you assign the downstream filter group. ■ <mac-address> – The MAC address (in xx:xx:xx:xx:xx:xx format) of the cable modem that provides network access to the CPE devices to which you assign the downstream filter group. ■ <sid> – The service identifier of the cable modem that provides network access to the CPE devices to which you assign the downstream filter group. Use the **show modem** command to determine the IP address, MAC address, and service identifier.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show cm-filter

row count: 1

Group Index Src Address      Src Mask         Dest Address     Dest Mask
----- ----- ---------------- ---------------- ---------------- -------------
    1     1 0.0.0.0          0.0.0.0          144.133.1.0      255.255.255.0
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem
row count: 11

MAC Address       IP Address      SID  CID   CPE   D:U Power  Timing Modem
                                                       (dbMV)        State
----------------- --------------- ---- ----  ---   --- ------ ------ -------
00:90:96:00:29:6d 201.1.1.103       67    1    0   1:2      0   2220 Registered
00:10:95:01:ef:d8 201.1.1.100       71    1    0   1:2      0   2209 Registered
00:10:95:04:0a:c4 201.1.1.102       63    1    0   1:2      0   2729 Registered
00:90:96:00:39:f9 201.1.1.101       68    1    0   1:2      0   2220 Registered
00:10:95:04:0a:b7 201.1.1.105       64    1    0   1:2      0   2728 Registered
00:90:96:00:29:71 201.1.1.104       66    1    0   1:2      0   2222 Registered
00:10:95:04:0a:bd 201.1.1.106       61    1    0   1:2      0   2729 Registered
00:10:95:04:0a:c3 0.0.0.0           62    0    0   1:2     -8   2729 Ranging

cli:192.168.208.3:interface:cable:csi(1/1/1)# cm modify cpe-downstream 1 71
```

Related Commands

- **cm-filter** on page 88
- **cm-filter-default cpe downstream** on page 96
- **cm modify active** on page 109
- **cm modify cm-downstream** on page 111
- **cm modify cm-upstream** on page 113
- **cm modify cpe-upstream** on page 117
- **cm modify learnable** on page 119
- **cm modify max-ip** on page 121
- **cm modify upstream** on page 123
- **cpe-control active** on page 130
- **show cm-filter** on page 389
- **show cm-filter-default** on page 390
- **show modem** on page 459

cm modify cpe-upstream

Assigns an upstream filter group to CPE devices that access the network through the specified cable modem. The value that this command specifies overrides the default that the **cm-filter-default cpe upstream** command specifies.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
cm modify cpe-upstream <group-id>
{<ip-address> | <mac-address> | <sid>}
```

Arguments

<group-id>	Specifies the ID of a filter group, which you create using the **cm-filter** command. Use the **show cm-filter** command to display available filter groups. Specifying a value of 0 disables upstream filtering for the CPE devices.
{<ip-address> | <mac-address> | <sid>}	Specifies one of the following values: ■ <ip-address> – The IP address of the cable modem that provides network access to the CPE devices to which you assign the upstream filter group. ■ <mac-address> – The MAC address (in xx:xx:xx:xx:xx:xx format) of the cable modem that provides network access to the CPE devices to which you assign the upstream filter group. ■ <sid> – The service identifier of the cable modem that provides network access to the CPE devices to which you assign the upstream filter group. Use the **show modem** command to determine the IP address, MAC address, and service identifier.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show cm-filter

row count: 1

Group Index Src Address      Src Mask         Dest Address     Dest Mask
----- ----- ---------------- ---------------- ---------------- -------------
    1     1 0.0.0.0          0.0.0.0          144.133.1.0      255.255.255.0
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem
row count: 11

MAC Address       IP Address      SID  CID   CPE   D:U Power  Timing Modem
                                                       (dbMV)        State
----------------- --------------- ---- ----  ---   --- ------ ------ -------
00:90:96:00:29:6d 201.1.1.103       67    1    0   1:2      0   2220 Registered
00:10:95:01:ef:d8 201.1.1.100       71    1    0   1:2      0   2209 Registered
00:10:95:04:0a:c4 201.1.1.102       63    1    0   1:2      0   2729 Registered
00:90:96:00:39:f9 201.1.1.101       68    1    0   1:2      0   2220 Registered
00:10:95:04:0a:b7 201.1.1.105       64    1    0   1:2      0   2728 Registered
00:90:96:00:29:71 201.1.1.104       66    1    0   1:2      0   2222 Registered
00:10:95:04:0a:bd 201.1.1.106       61    1    0   1:2      0   2729 Registered
00:10:95:04:0a:c3 0.0.0.0           62    0    0   1:2     -8   2729 Ranging

cli:192.168.208.3:interface:cable:csi(1/1/1)# cm modify cpe-upstream 1 71
```

Related Commands

- **cm-filter** on page 88
- **cm-filter-default cpe upstream** on page 98
- **cm modify active** on page 109
- **cm modify cm-downstream** on page 111
- **cm modify cm-upstream** on page 113
- **cm modify cpe-upstream** on page 117
- **cm modify learnable** on page 119
- **cm modify max-ip** on page 121
- **cm modify upstream** on page 123
- **cpe-control active** on page 130
- **show cm-filter** on page 389
- **show cm-filter-default** on page 390
- **show modem** on page 459

cm modify learnable

Enables the ability of the CMTS to discover CPE IP addresses associated with the specified cable modem. The value that this command specifies overrides the default that the **cpe-control learnable** command specifies.

Use the no form of the command to disable the ability of the CMTS to discover CPE IP addresses associated with the specified cable modem.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
cm modify learnable {<ip-address> | <mac-address> | <sid>}

no cm modify learnable {<ip-address> | <mac-address> | <sid>}
```

Arguments

{<ip-address> | <mac-address> | <sid>}	Specifies one of the following values: ■ <ip-address> – The IP address of the cable modem for which you want to enable or disable the ability of the CMTS to discover IP addresses. ■ <mac-address> – The MAC address (in xx:xx:xx:xx:xx:xx format) of the cable modem for which you want to enable or disable the ability of the CMTS to discover IP addresses. ■ <sid> – The service identifier of the cable modem for which you want to enable or disable the ability of the CMTS to discover IP addresses. Use the **show modem** command to determine the IP address, MAC address, and service identifier.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem

row count: 11

MAC Address       IP Address      SID  CID   CPE   D:U Power  Timing Modem
                                                       (dbMV)        State
----------------- --------------- ---- ----  ---   --- ------ ------ -------
00:90:96:00:29:6d 201.1.1.103      67    1    0   1:2      0   2220 Registered
00:10:95:01:ef:d8 201.1.1.100      71    1    0   1:2      0   2209 Registered
00:10:95:04:0a:c4 201.1.1.102      63    1    0   1:2      0   2729 Registered
00:90:96:00:39:f9 201.1.1.101      68    1    0   1:2      0   2220 Registered
00:10:95:04:0a:b7 201.1.1.105      64    1    0   1:2      0   2728 Registered
00:90:96:00:29:71 201.1.1.104      66    1    0   1:2      0   2222 Registered
00:10:95:04:0a:bd 201.1.1.106      61    1    0   1:2      0   2729 Registered
00:10:95:01:f0:05 201.1.1.107      75    1    0   1:2      0   2212 Registered
00:90:83:36:82:f1 201.1.1.110      73    1    0   1:2      0   1233 Registered
00:90:83:32:9f:8c 201.1.1.111      70    1    0   1:2      0   1655 Registered
00:10:95:04:0a:c3 0.0.0.0           62    0    0   1:2     -8   2729 Ranging

cli:192.168.208.3:interface:cable:csi(1/1/1)# cm modify learnable 71
```

Related Commands

- **cm modify cm-downstream** on page 111
- **cm modify cm-upstream** on page 113
- **cm modify cpe-downstream** on page 115
- **cm modify cpe-upstream** on page 117
- **cm modify max-ip** on page 121
- **cm modify upstream** on page 123
- **cpe-control learnable** on page 131
- **show modem** on page 459

cm modify max-ip

Sets the maximum number of IP addresses of CPE devices that can access the network through the specified cable modem. The value that this command specifies overrides the default that the **cpe-control max-ip** command specifies.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
cm modify max-ip <number> {<ip-address> | <mac-address> | <sid>}
```

Arguments

<number>	Specifies the maximum number of IP addresses of CPE devices that can access the network through the specified cable modem. Values range from 0 (no CPE allowed) to 16. The default value is the value set by the **cpe-control max-ip** command.
{<ip-address> | <mac-address> | <sid>}	Specifies one of the following values: ■ <ip-address> – The IP address of the cable modem that provides network access to CPE. ■ <mac-address> – The MAC address (in xx:xx:xx:xx:xx:xx format) of the cable modem that provides network access to CPE. ■ <sid> – The service identifier of the cable modem that provides network access to CPE. Use the **show modem** command to determine the IP address, MAC address, and service identifier of the cable modem.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem

row count: 12

MAC Address       IP Address      SID  CID  CPE D:U Power  Timing Modem
                                                   (dbMV)        State
----------------- --------------- ---- ---- --- --- ------ ------ ----------
00:90:96:00:29:71 201.1.1.112       1    1   0 1:2      0   2216 Registered
00:90:83:36:82:ee 201.1.1.108       2    0   0 1:2      0   1233 RegBpiTek
00:90:96:00:29:6d 201.1.1.104       3    1   0 1:2      0   2217 Registered
00:10:95:04:0a:b7 201.1.1.109       4    1   0 1:2      0   2725 Registered
00:90:96:00:39:f9 201.1.1.101       5    1   0 1:2      0   2212 Registered
00:10:95:01:ef:d8 201.1.1.100       6    1   0 1:2      0   9995 Registered
00:10:95:04:0a:c3 0.0.0.0           7   57   0 1:2     -9   2726 Ranging
00:90:96:00:39:7f 201.1.1.102       8    1   0 1:2      0   2219 Registered
00:10:95:04:0a:c4 201.1.1.110       9    1   0 1:2      0   2724 Registered
00:10:95:01:f0:05 201.1.1.103      10    1   0 1:2      0  26614 Registered
00:90:83:32:9f:8c 201.1.1.106      11    1   0 1:2      0   1639 Registered
00:90:83:36:82:f1 201.1.1.107      12    1   0 1:2      0   1239 Registered

cli:192.168.208.3:interface:cable:csi(1/1/1)# cm modify max-ip 16 2
```

Related Commands

- **cm modify cm-downstream** on page 111
- **cm modify cm-upstream** on page 113
- **cm modify cpe-downstream** on page 115
- **cm modify cpe-upstream** on page 117
- **cm modify learnable** on page 119
- **cm modify upstream** on page 123
- **cpe-control max-ip** on page 132
- **show modem** on page 459

cm modify upstream

Moves the specified cable modem to a new upstream port. Note that the new upstream port must be enabled (up).

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
cm modify upstream <new-upstream-channel>
{<ip-address> | <mac-address> | <sid>}
```

Arguments

<new-upstream-channel>	Specifies the upstream channel number. Values range from 1 to 4 for 1x4 modules and 1 to 6 for 1x6 modules.
{<ip-address> | <mac-address> | <sid>}	Specifies one of the following values: ■ <ip-address> – The IP address of the cable modem that you want to reassign to the specified upstream port. ■ <mac-address> – The MAC address (in xx:xx:xx:xx:xx:xx format) of the cable modem that you want to reassign to the specified upstream port. ■ <sid> – The service identifier of the cable modem that you want to reassign to the specified upstream port. Use the **show modem** command to determine the IP address, MAC address, and service identifier of the cable modem.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem

row count: 11

MAC Address       IP Address      SID  CID   CPE   D:U Power  Timing Modem
                                                       (dbMV)        State
----------------- --------------- ---- ----  ---   --- ------ ------ -------
00:90:96:00:29:6d 201.1.1.103      67    1    0   1:1      0   2220 Registered
00:10:95:01:ef:d8 201.1.1.100      71    1    0   1:1      0   2208 Registered
00:10:95:04:0a:c4 201.1.1.102      63    1    0   1:1      0   2729 Registered
00:90:96:00:39:f9 201.1.1.101      68    1    0   1:1      0   2220 Registered
00:10:95:04:0a:b7 201.1.1.105      64    1    0   1:1      0   2728 Registered
00:90:96:00:29:71 201.1.1.104      66    1    0   1:1      0   2223 Registered
00:10:95:04:0a:bd 201.1.1.106      61    1    0   1:1      0   2729 Registered
00:10:95:01:f0:05 201.1.1.107      75    1    0   1:1      0   2214 Registered
00:90:83:36:82:f1 201.1.1.110      73    1    0   1:1      0   1235 Registered
00:90:83:32:9f:8c 201.1.1.111      70    1    0   1:1      0   1656 Registered
00:10:95:04:0a:c3 0.0.0.0           62    0    0   1:1     -8   2729 Ranging

cli:192.168.208.3:interface:cable:csi(1/1/1)# cm modify upstream 2 67
Modifying Cable Modem with SID: 67 to upstream channel: 2
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem

row count: 11

MAC Address       IP Address      SID  CID   CPE   D:U Power  Timing Modem
                                                       (dbMV)        State
----------------- --------------- ---- ----  ---   --- ------ ------ -------
00:90:96:00:29:6d 201.1.1.103      67    1    0   1:2      0   2220 Registered
00:10:95:01:ef:d8 201.1.1.100      71    1    0   1:1      0   2208 Registered
00:10:95:04:0a:c4 201.1.1.102      63    1    0   1:1      0   2729 Registered
00:90:96:00:39:f9 201.1.1.101      68    1    0   1:1      0   2220 Registered
00:10:95:04:0a:b7 201.1.1.105      64    1    0   1:1      0   2728 Registered
00:90:96:00:29:71 201.1.1.104      66    1    0   1:1      0   2223 Registered
00:10:95:04:0a:bd 201.1.1.106      61    1    0   1:1      0   2729 Registered
00:10:95:01:f0:05 201.1.1.107      75    1    0   1:1      0   2214 Registered
00:90:83:36:82:f1 201.1.1.110      73    1    0   1:1      0   1235 Registered
00:90:83:32:9f:8c 201.1.1.111      70    1    0   1:1      0   1656 Registered
00:10:95:04:0a:c3 0.0.0.0           62    0    0   1:1     -8   2729 Ranging
```

Related Commands

- **show modem** on page 459

cm reset

Resets a cable modem. You can specify the modem that you want to reset in terms of its IP address, MAC address, or Service Identifier (SID).

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

To reset a single modem:

```
cm reset {<ip-address> | <mac-address> | <sid>}
```

To reset multiple modems with common hexidecimal MAC address values:

```
cm reset <hex-values> match
```

To reset all modems attached to a selected network:

```
cm reset <address-string> match
```

Arguments

{<ip-address> | <mac-address> | <sid>}	Specifies one of the following values: ■ <ip-address> – The IP address of the cable modem that you want to reset. ■ <mac-address> – MAC address (in xx:xx:xx:xx:xx:xx format) of the cable modem that you want to reset. This is especially useful in resetting modems that have not yet received an IP address. ■ <sid> – The service identifier of the cable modem that you want to reset. Use the **show modem** command to determine the IP address, MAC address, and service identifier of the cable modem.
<hex-values>	Specifies the hex values that you want to match, which you enter with "ff" values specified as wildcards. For example, if you want to reset all cable modems from vendor 00:50:72, enter the following command: **cm reset 00:50:72:ff:ff:ff match**

<address-string>	Specifies the IP address that you want to match, which you enter with a "255" wildcard mask. For example, if you want to reset all cable modems attached to subnet 189.23.3.x, enter the following command: **cm reset 189.23.3.255 match**
match	Keyword used to define an address or MAC address match.

Example 1

The following example resets a single modem with the SID of 71.

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem

row count: 12

MAC Address       IP Address      SID  CID   CPE   D:U Power  Timing Modem
                                                      (dbMV)        State
----------------- --------------- ---- ----  ---   --- ------ ------ -------
00:90:96:00:29:6d 201.1.1.103     453    1    0   1:2      0   2219 Registered
00:10:95:01:ef:d8 201.1.1.100      71    1    0   1:2      0   2209 Registered
00:10:95:04:0a:c4 201.1.1.102      63    1    0   1:2      0   2730 Registered
00:90:96:00:39:f9 201.1.1.101      68    1    0   1:2      0   2219 Registered
00:10:95:04:0a:b7 201.1.1.105      64    1    0   1:2      0   2728 Registered
00:90:96:00:29:71 201.1.1.104      66    1    0   1:2      0   2223 Registered
00:10:95:04:0a:bd 201.1.1.106      61    1    0   1:2      0   2728 Registered
00:10:95:01:f0:05 201.1.1.107      75    1    0   1:2      0   2213 Registered
00:90:96:00:39:7f 0.0.0.0         647    0    0   1:2      0   2219 RangComple
00:90:83:36:82:f1 201.1.1.110      73    1    0   1:2      0   1235 Registered
00:90:83:32:9f:8c 201.1.1.111      70    1    0   1:2      0   1656 Registered
00:10:95:04:0a:c3 0.0.0.0           62    0    0   1:2     -9   2730 Ranging

cli:192.168.208.3:interface:cable:csi(1/1/1)# cm reset 71
Resetting Cable Modem with SID: 71
```

Example 2

The following example displays the modems attached to cable interface 1/1/1 then uses the match argument to reset all modems with the vendor ID: 00:90:96.

```
cli:interface:cable:csi(1/1/1)# show modem

row count: 11

MAC Address       IP Address      SID  CID   CPE   D:U Power  Timing Modem
                                                      (dbMV)        State
----------------- --------------- ---- ----  ---   --- ------ ------ -------
00:90:96:00:29:6d 201.1.1.103      453    1    0   1:2      0   2217 Registered
00:10:95:01:ef:d8 201.1.1.100      654    1    0   1:2      0   2209 Registered
00:10:95:04:0a:c4 201.1.1.102       63    1    0   1:2      0   2729 Registered
00:90:96:00:39:f9 201.1.1.101       68    1    0   1:2      0   2220 Registered
00:10:95:04:0a:b7 201.1.1.105       64    1    0   1:2      0   2727 Registered
00:90:96:00:29:71 201.1.1.104       66    1    0   1:2      0   2222 Registered
00:10:95:04:0a:bd 201.1.1.106       61    1    0   1:2     -2   2729 Registered
00:10:95:01:f0:05 201.1.1.107       75    1    0   1:2      0   2214 Registered
00:90:83:36:82:f1 201.1.1.110       73    1    0   1:2      0   1234 Registered
00:90:83:32:9f:8c 201.1.1.111       70    1    0   1:2      0   1657 Registered
00:10:95:04:0a:c3 0.0.0.0           62    0    0   1:2     -8   2730 Ranging

cli:interface:cable:csi(1/1/1)# cm reset 00:90:96:ff:ff:ff match
Resetting Cable Modem with Mac Address: 00:90:96:00:29:6d
Resetting Cable Modem with Mac Address: 00:90:96:00:39:f9
Resetting Cable Modem with Mac Address: 00:90:96:00:29:71
```

Example 3

The following example uses the match argument against the IP address parameter to reset all cable modems on the 201.1.1.0 subnet.

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem

row count: 9

MAC Address       IP Address      SID  CID   CPE   D:U Power  Timing Modem
                                                       (dbMV)        State
----------------- --------------- ---- ----  ---   --- ------ ------ -------
00:10:95:01:ef:d8 201.1.1.100      654    1    0   1:2      0   2208 Registered
00:10:95:04:0a:c4 201.1.1.102       63    1    0   1:2      0   2729 Registered
00:10:95:04:0a:b7 201.1.1.105       64    1    0   1:2      0   2727 Registered
00:10:95:04:0a:bd 201.1.1.106       61    1    0   1:2      0   2729 Registered
00:10:95:01:f0:05 201.1.1.107       75    1    0   1:2      0   2213 Registered
00:90:96:00:39:7f 201.1.1.108      659    0    0   1:2      0   2217 Registered
00:90:83:36:82:f1 201.1.1.110       73    1    0   1:2      0   1234 Registered
00:90:83:32:9f:8c 201.1.1.111       70    1    0   1:2      0   1657 Registered
00:10:95:04:0a:c3 0.0.0.0           62    0    0   1:2     -8   2730 Ranging

cli:192.168.208.3:interface:cable:csi(1/1/1)# cm reset 201.1.1.255 match
Resetting Cable Modem with Ip Address: 201.1.1.100
Resetting Cable Modem with Ip Address: 201.1.1.102
Resetting Cable Modem with Ip Address: 201.1.1.105
Resetting Cable Modem with Ip Address: 201.1.1.106
Resetting Cable Modem with Ip Address: 201.1.1.107
Resetting Cable Modem with Ip Address: 201.1.1.108
Resetting Cable Modem with Ip Address: 201.1.1.110
Resetting Cable Modem with Ip Address: 201.1.1.111
```

Related Commands

- **show modem** on page 459

connect

Connects you to a remote Cuda 12000 IP Access Switch.

Mode

`root`

Syntax

`connect` <ip-address> [**`password`** <password>] [**`user`** <username>]

Arguments

<ip-address>	Specifies the IP address of the remote Cuda 12000.
password <password>	Specifies the password for accessing the remote Cuda 12000. No username and password are required if the same username/password combination enables you to access both the local Cuda 12000 and the remote Cuda 12000.
user <username>	Specifies the username for accessing the remote Cuda 12000. No username and password are required if the same username/password combination enables you to access both the local Cuda 12000 and the remote Cuda 12000.

Example

```
cli:192.165.204.5:root# connect 192.165.204.30 password foo user root
Connecting to host: 192.165.204.30 as user root
Connecting to 192.165.204.30...
Java Server version is compatible
logon complete
cli:192.165.204.30:root#
```

cpe-control active

Specifies whether the CMTS performs subscriber management (such as filtering and enforcing CPE address limits) by default for cable modems. You can use the **cm modify active** command to override this default setting on a modem-by-modem basis.

Use the no form of the command to disable subscriber management.

Mode

Any.

Syntax

cpe-control active

no cpe-control active

Example

```
cli:interface:cable:csi(1/1/1)# cpe-control active
cli:interface:cable:csi(1/1/1)# show cpe-control
MAX IP                                    16
Active                                  True
Learnable                               True
```

Related Commands

- **cm modify active** on page 109
- **cpe-control learnable** on page 131
- **cpe-control max-ip** on page 132
- **show cpe-control** on page 395
- **show interface cable <c/s/i> cpe-control** on page 416

cpe-control learnable

Specifies whether the Cuda 12000 discovers CPE IP addresses associated with cable modems on all CMTS interfaces by default. You can use the **cm modify learnable** command to override this default setting on a modem-by-modem basis.

Use the no form of the command to disable the ability to discover CPE IP addresses.

Mode

Any.

Syntax

```
cpe-control learnable

no cpe-control learnable
```

Example

```
cli:interface:cable:csi(1/1/1)# cpe-control learnable
value = true
cli:interface:cable:csi(1/1/1)# show cpe-control
MAX IP                                    16
Active                                 False
Learnable                               True
```

Related Commands

- **cm modify learnable** on page 119
- **cpe-control active** on page 130
- **cpe-control max-ip** on page 132
- **show cpe-control** on page 395
- **show interface cable <c/s/i> cpe-control** on page 416

cpe-control max-ip

Specifies the default maximum number of CPE IP addresses that can access the network through a single cable modem. You can use the **cm modify max-ip** command to override this default setting on a modem-by-modem basis.

Use the no form of the command to set the maximum number of CPE IP addresses to zero.

Mode

Any.

Syntax

cpe-control max-ip <number>

no cpe-control max-ip

Arguments

<number>	Specifies the default maximum number of IP addresses of CPE devices that can access the network through a single cable modem. Values range from 0 (no CPE devices allowed) to 16 (the default).

Example

```
cli:interface:cable:csi(1/1/1)# cpe-control max-ip 16
cli:interface:cable:csi(1/1/1)# show cpe-control
MAX IP                                  16
Active                                False
Learnable                              True
```

Related Commands

- **cm modify max-ip** on page 121
- **cpe-control active** on page 130
- **cpe-control learnable** on page 131
- **show cpe-control** on page 395
- **show interface cable <c/s/i> cpe-control** on page 416

cpu-utilization

Enables the CPU utilization feature on each module. After you enable CPU utilization using this command, you can use the **show cpu-utilization** feature to display CPU usage for module processors.



***Use the cpu-utilization and no cpu-utilization commands to toggle the CPU usage feature* on *and* off.**

Mode

```
slot(<c/s>)
```

Syntax

cpu-utilization

Example

```
cli:172.16.19.10:slot(1/1)# no cpu-utilization
cli:172.16.19.10:slot(1/1)# show cpu-utilization
CPU Utilization is turned off
cli:172.16.19.10:slot(1/1)# cpu-utilization
cli:172.16.19.10:slot(1/1)# show cpu-utilization

Cpu Utilization, Duration: 1 Hour in 1 minute intervals

row count: 120

CPU Id Time       Avg CPU    Max CPU    Max CPU
       (Minutes)  Usage      Usage      Usage Time
------ ---------- ---------- ---------- ----------
     1          1          0          0          0
     1          2          0          0          0
     1          3          0          0          0
     1          4          0          0          0
     1          5          0          0          0
     1          6          0          0          0
     1          7          0          0          0
     1          8          0          0          0
     1          9          0          0          0
     1         10          0          0          0
--More--
```

crc

Configures cyclic redundancy checking (CRC) on the current POS interface. CRC is an error-checking mechanism that utilizes a calculated numeric value to detect errors in data transmission. The sending device calculates a frame check sequence (FCS), then appends the value to outgoing packets. The receiving device then recalculates the FCS and verifies that it amounts to the same value calculated by the sender. If not, it assumes the packet transmission is in error and requests that the sender retransmit the packet.

POS interfaces use 32-bit CRC by default. Note that both the sender and receiver must be configured to use the same CRC — either 16-bit or 32-bit — error checking. Use the **no crc** command to configure the interface to use the default 32-bit CRC.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
crc {16 | 32}
```

Arguments

16	Configures the interface to use 16-bit CRC error checking.
32	Configures the interface to use 32-bit CRC error checking. (default)

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# crc 16
cli#
```

Related Commands

- **interface** on page 181
- **no crc** on page 266
- **show interface pos** on page 437

db-check

Validates the LDAP database access information contained in the provision.cfg file. Use this command to ensure the system can communicate with the provisioning database.

Mode

`root`

Syntax

`db-check`

Example

```
cli:172.16.19.10:root# db-check
LDAP database accessible
cli:172.16.19.10:root#
```

Related Commands

- **db-connect** on page 136
- **show db-connect** on page 398

db-connect

Configures the provisioning server ID, IP address of the host where the LDAP database resides, and other information required to access the LDAP database. This command tests the connection to the LDAP database and, if successful, creates a provision.cfg file containing the access information that you specified. The provision.cfg file resides in /bas/data/provision on the management module and is referred to whenever a provisioning application has to access the database.

Mode

`root`

Syntax

Note that you must pass the command line arguments to the command as a single string in the order specified below and as shown in the example:

```
dbconnect <ldap-server-object-name> <ldap-server-ip-address>
<ldap-server-port-number> <ldap-server-username>
<ldap-server-password>
```

Arguments

<ldap-server-object-name>	Provisioning server ID of the provisioning server. If you are using the integrated provisioning database on the local host, this value is 1.
<ldap-server-ip-address>	IP address of the host where the LDAP database resides. If you are using the integrated provisioning database on the local host, this value is 127.0.0.1.
<ldap-server-port-number>	TCP port used to access the LDAP database. By default, port 389 is used to access LDAP databases.
<ldap-server-username>	Username required to access the LDAP database. The default username for the integrated provisioning server is cn=Directory Manager.
<ldap-server-password>	Password required to access the LDAP database. The default password for the integrated LDAP database is "Directory Manager."

Example

```
cli:root# db-connect 1 127.0.0.1 389 "cn=Directory Manager, o=basystems.com"
bas_ldap
Verifying parameters...
LDAP database accessible
Successfully updated config file
cli:root#
```

Related Commands

- **db-check** on page 135

del arp

Deletes the specified static ARP entry from the ARP cache of the current Ethernet interface. Note that static ARP entries are supported on Ethernet interfaces only.

Mode

```
interface:ethernet:csi(<c/s/i>)
```

Syntax

```
del arp <ip-address>
```

Arguments

<ip-address>	Address that you want to remove from the ARP cache of the current interface.

Example

The following example deletes the 192.168.111.6 static ARP address from Ethernet interface 1/11/1:

```
cli:172.16.19.10:root# interface 1/11/1
mode: interface:ethernet:csi(1/11/1)
cli:172.16.19.10:interface:ethernet:csi(1/11/1)# show arp

row count: 1

IP Address       MAC Address        Type
---------------- ------------------ ----------
192.168.111.6    a0:41:de:78:ac:9e      static

cli:172.16.19.10:interface:ethernet:csi(1/11/1)# del arp 192.168.111.6
cli:172.16.19.10:interface:ethernet:csi(1/11/1)# show arp


IP Address       MAC Address        Type
---------------- ------------------ ----------
```

Related Commands

- **add arp** on page 50
- **show arp** on page 374

dhcp-authority

Use this command to configure Dynamic Host Configuration Protocol (DHCP) authority on the current interface. You can both enable and disable DHCP authority using this command, as well as configure authority ranges.

DHCP authority is a security feature that prevents spoofing (unauthorized use) of DHCP assigned IP addresses. DHCP authority provides this security by tagging all Address Resolution Protocol (ARP) entries that consist of IP addresses that fall within the specified range.

This feature is termed DHCP Authority because those tagged as being assigned via DHCP take precedence over dynamically assigned (non-DHCP tagged) ARP entries. When viewing the ARP cache, those entries that are protected by DHCP authority are labelled as type other. This means that only that specific MAC address will be allowed to map to that specific IP address; ARP requests by other MAC addresses for that IP address are dropped.

Configuration of DHCP authority consists of a three-step process:

1. Enable DHCP authority on the selected interface.
2. Configure a DHCP authority range specific to the IP interface on the select interface.
3. Reboot the hosts *(modems, CPE devices)* whose IP addresses you want to protect so that the DHCP authority configuration takes effect. Note that the ARP entries are tagged as DHCP assigned (indicated by type *other*) upon DHCP acknowledgement of the allocated IP address.

Use the **no dhcp-authority** command to remove an authority range from the current interface. For example, **no dhcp authority 1** removes the DHCP authority range 1 from the interface.

Mode

```
interface:<type>:csi(<c/s/i>)
```

Syntax

To enable or disable DHCP authority on the current interface:

dhcp-authority {enable | disable}

To configure DHCP authority ranges on the current interface:

dhcp-authority <number> **start** <start-ip-address> **end** <end-ip-address>

To remove a DHCP authority range from the current interface:

no dhcp-authority <number>

Arguments

enable	Enable DHCP authority on the current interface.
disable	Disable DHCP authority on the current interface.
<number>	Index number identifying the authority range on the current interface. Note that you cannot modify ranges; you must delete the range and then redefine it.
<start-ip-address>	First IP address in the DHCP authority range.
<end-ip-address>	Last IP address in the DHCP authority range.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show dhcp-authority


Range Number     Lower Range      Upper Range      Status
---------------- ---------------- ---------------- ------

DHCP Authority Status                disable

cli:192.168.208.3:interface:cable:csi(1/1/1)# dhcp-authority 4 start
201.1.2.140 end 201.1.2.160
cli:192.168.208.3:interface:cable:csi(1/1/1)# show dhcp-authority

row count: 1

Range Number     Lower Range      Upper Range      Status
---------------- ---------------- ---------------- ------
               4 201.1.2.140      201.1.2.160           1

DHCP Authority Status                disable

cli:192.168.208.3:interface:cable:csi(1/1/1)# dhcp-authority enable
cli:192.168.208.3:interface:cable:csi(1/1/1)# show dhcp-authority

row count: 1

Range Number     Lower Range      Upper Range      Status
---------------- ---------------- ---------------- ------
               4 201.1.2.140      201.1.2.160           1

DHCP Authority Status                 enable
```

Related Commands

- **show dhcp-authority** on page 399

dhcp-policy

Defines Dynamic Host Configuration Protocol (DHCP) policies. DHCP Policies allow you to control and restrict the forwarding of DHCP requests. These policies allow matching on several parameters in the DHCP packet, then use the result of this matching to determine which list of servers to forward the packet to; or it can reject (drop) the packet to deny the requesting client an address.

You use the **no dhcp-policy** command to remove the configuration.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
dhcp-policy {<policy-number> | default} {deny | permit}
  {<ip-address> ... | forward-internal [disable]}
  [agent-option {cm | cpe} | cmmac <mac-address> | interface <c/s/i> |
  mac <mac-address> [mask <mask>]] [vendor-class-id {cm | mta}]
  [description <string>]

no dhcp-policy {<policy-number> | all}
```

Arguments

{<policy-number> | **default**}	Specifies one of the following values: ■ <number> – Number that you want to assign to this policy. Valid range is 1 to 1000. ■ default – Default policy.
{**deny** | **permit**}	Specifies one of the following values: ■ deny – Creates a policy that denies matching packets. ■ permit – Creates a policy that allows matching packets to pass.

{<ip-address> **...** | **forward-internal** [**disable**]}	Specifies one of the following values: ■ <ip-address>... – A list of IP addresses to which you want the current cable interface to forward DHCP packets. ■ forward-internal – Specifies that the current cable interface forwards DHCP requests internally (that is, to a DHCP server on the local Cuda 12000). Optionally, you can specify the **disable** keyword to disable internal forwarding.
agent-option {**cm** | **cpe**}	Specifies one of the following values: ■ cm – Specifies that cable modem DHCP requests are matched. ■ cpe – Specifies that CPE DHCP requests are matched.
cmmac <mac address>	Specifies that DHCP requests originating from the specified cable modem MAC address are matched.
interface <c/s/i>	Specifies that any DHCP requests arriving on the specified physical interface are matched.
mac <mac address> [**mask** <mask>]	Specifies a keyword/address combination to match on the source MAC address of a DHCP request. Optionally, you can specify a mask to wildcard the source mac address that you want to match with a combination of 0s and Fs. For example, the following MAC address/mask pair would match all addresses starting with 00:02:09: MAC address: 00:02:09:00:00:00 Mask: FF:FF:FF:00:00:00
vendor-class-id {**cm** | **mta**}	Specifies one of the following values: cm – Specifies that DHCP requests that contain a cable modem vendor class are matched. mta – Specifies that DHCP requests that contain an MTA vendor class are matched.
description <string>	Specifies an alphanumeric text string describing this policy.
all (no form of the command only)	Deletes all policies.

Example

The following example configures cable interface 1/1/1 to forward all DHCP requests from CPE devices to server 10.1.13.5:

```
cli:interface:cable:csi(1/1/1)# dhcp-policy 3 permit 10.1.13.5 agent-option
cpe
```

Related Commands

- **show dhcp-policy** on page 401

dhcp-relay

Configures DHCP parameters on the current cable (CMTS) interface. DHCP is used within a DOCSIS-compliant network to allocate IP addresses and configure modems with other IP parameters. DOCSIS modules function as DHCP relay agents and forward DHCP requests from modems and connected CPE devices to a DHCP server. Use this command to specify the DHCP server that you want the cable interface to use, as well as configure other DHCP relay parameters, such as gateways for cable modems, CPE devices, and MTAs.

Use the no form of the command to remove a gateway or server.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
dhcp-relay {enable | disable}
[cm-gateway <gi-address>]
[cpe-gateway <gi-address>]
[add-agent-options {enable | disable}]
[drop-mismatch {enable | disable}]
[relay-mode {append | replace | untouch | discard}]
[max-pkt-len <number>]
[mta-gateway <gi-address>]
[server <gi-address>]

no dhcp-relay {cm-gateway | cpe-gateway | mta-gateway | server}
```

Arguments

{**enable** | **disable**}	Specifies one of the following values: ■ enable – Enables DHCP on the current cable interface. ■ disable – Disables DHCP on the current cable interface.
cpe-gateway <gi-address>	Configures the IP gateway for CPE devices connected to the current interface. You must enter one of the IP addresses (interfaces) that you have added to the current physical interface. The relay agent on the card places this address in the giaddr field of the DHCP packet before forwarding the requests to the DHCP server. The server then checks the LDAP database to see if you have configured subnets for this network, and then selects the network address based on the subnet definition. Note that you must define subnets for the address that you specify in this field; if the DHCP server cannot find a corresponding subnet for this network, it will not assign IP addresses.
cm-gateway <gi-address>	Configures the IP gateway for cable modems connected to the current interface. You must enter one of the IP addresses (interfaces) that you have added to the current physical interface. The relay agent on the card places this address in the giaddr field of the DHCP packet before forwarding the requests to the DHCP server. The server then checks the LDAP database to see if you have configured subnets for this network, and then selects the network address based on the subnet definition. Note that you must define subnets for the address that you specify in this field; if the DHCP server cannot find a corresponding subnet for this network, it will not assign the requesting device an IP address.
add-agent-options {**enable** | **disable**}	Add additional DHCP relay agent functionality. Enable this parameter if you have specified two different IP gateways – one for modems and a different one for CPE devices.
drop-mismatch {**enable** | **disable**}	Used to authenticate DHCP client-to -server communication. If enabled, configures the system to drop DHCP server replies that do not contain an Agent ID suboption that identifies origination from the DHCP server.

relay-mode {**append** | **replace** | **untouch** | **discard**}	Set the relay mode used on this cable interface.
max-pkt-len <number>	Sets the DHCP relay maximum packet length. Values range from 576 to 1518.
mta-gateway <gi-address>	Sets the IP address of the MTA gateway.
server <gi-address>	Sets the IP address of the DHCP server.

Example

```
cli:interface:cable:csi(1/1/1)# dhcp-relay add-agent-options enable
cm-gateway 201.1.1.1 cpe-gateway 201.1.2.1
```

Related Commands

- **show dhcp-relay** on page 402

downstream frequency

Sets the downstream frequency on a selected CMTS card. The downstream forwarding slot must be active when setting the downstream frequency.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

downstream frequency <freq-number>

Arguments

<freq-number>	Frequency to which you want to set the downstream ports on the selected card, in MHz.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# downstream frequency 507
cli:172.16.19.10:interface:cable:csi(1/1/1)# show downstream
C/S/P                          1 / 1 / 2 / 2
Frequency                              507.0  (MHz)
Interleave                 taps32Increment4
Modulation                            qam256
ChannelWidth                               6  (MHz)
ChannelPower                             550  (1/10 dBmV)
AnnexType                              Annex B
Symbol Rate                            5360537  (baud)
Admin Status                              up
Operational Status                        up
Out Octets                         732210940
Out Unicast Packets                    16236
Out Multicacast Packets            16453680
Out Broadcast Packets                    407
Out Errors                                 0
Out Discards                               0
```

Related Commands

- **downstream interleave-depth** on page 150
- **downstream modulation** on page 151
- **downstream no shutdown** on page 153
- **downstream shutdown** on page 154
- **downstream transmit-power** on page 155
- **show interface cable <c/s/i> downstream** on page 417
- **show downstream** on page 403

downstream interleave-depth

Sets the interleave-depth used for downstream ports on a selected CMTS card.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
downstream interleave-depth <number>
```

Arguments

<number>	Interleave depth that you want the CMTS card to use: ■ DOCSIS Acceptable values: **8** |**16** | **32** | **64** |**128** ■ EuroDOCSIS: Acceptable value: **12**

Example

The following example configures the CMTS card in slot 1 of chassis 1 to use an interleave depth of 128.

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# downstream interleave-depth 128
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **downstream frequency** on page 148
- **downstream modulation** on page 151
- **downstream no shutdown** on page 153
- **downstream shutdown** on page 154
- **downstream transmit-power** on page 155
- **show interface cable <c/s/i> downstream** on page 417
- **show downstream** on page 403

downstream modulation

Sets the downstream modulation scheme used on the selected CMTS interface.



Caution: Specifying a new downstream modulation causes the module to save the new setting to FLASH and reboot. If other CMTS parameters have been changed, you should use the save command to persist those changes before issuing the downstream modulation change.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
downstream modulation {qam64 | qam256}
```

Arguments

qam64	Sets downstream modulation on the selected CMTS card to 64 QAM.
qam256	Sets downstream modulation on the selected CMTS card to 256 QAM.

Example

The following example sets downstream modulation for the CMTS card in slot 1, chassis 1 to 256 QAM.

```
cli:172.16.19.10:root# save
Saving slot: 1/1 ..
Save request passed for slot 1/1
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# downstream modulation qam256
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **downstream frequency** on page 148
- **downstream interleave-depth** on page 150
- **downstream no shutdown** on page 153
- **downstream shutdown** on page 154
- **downstream transmit-power** on page 155
- **show interface cable <c/s/i> downstream** on page 417
- **show downstream** on page 403

downstream no shutdown

Sets the downstream channel status on the selected CMTS card to *up*.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
downstream no shutdown
```

Example

The following example sets the channel status on the CMTS card in slot 1, chassis 1 to the up state.

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# downstream no shutdown
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **downstream frequency** on page 148
- **downstream interleave-depth** on page 150
- **downstream modulation** on page 151
- **downstream shutdown** on page 154
- **downstream transmit-power** on page 155
- **show interface cable <c/s/i> downstream** on page 417
- **show downstream** on page 403

downstream shutdown

Sets the downstream channel status on the selected CMTS card to *down*.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
downstream shutdown
```

Example

The following example sets the channel status on the CMTS card in slot 1, chassis 1 to the down state.

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# downstream shutdown
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **downstream frequency** on page 148
- **downstream interleave-depth** on page 150
- **downstream modulation** on page 151
- **downstream no shutdown** on page 153
- **downstream transmit-power** on page 155
- **show interface cable <c/s/i> downstream** on page 417
- **show downstream** on page 403

downstream transmit-power

Configures the downstream transmit power on the selected CMTS card.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
downstream transmit-power <number>
```

Arguments

<number>	Number to which to set the downstream transmit power for the current CMTS card. Acceptable values: 0 – 650. The value you specify is then divided by ten. For example, if you specify 550, then the real value is 55.0 dBmV.

Example

The following example sets the downstream transmit power to 55.0 dBmV.

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# downstream transmit-power 550
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **downstream frequency** on page 148
- **downstream interleave-depth** on page 150
- **downstream modulation** on page 151
- **downstream no shutdown** on page 153
- **downstream shutdown** on page 154
- **show interface cable <c/s/i> downstream** on page 417
- **show downstream** on page 403

duplex

Sets the duplex mode on an Ethernet port.

Mode

```
interface:ethernet:csi(<c/s/i>)
```

Syntax

duplex {**auto** | **full** | **half**}

Arguments

{**auto** | **full** | **half**}	Specifies one of the following values: ■ auto – Duplex mode is automatically configured. Note that, if you configure the port to automatically negotiate duplex mode, you also set the port to automatically negotiate speed. ■ full – Sets the port to full-duplex mode (port can send and receive simultaneously). Note that, if you configure the port for full duplex or half duplex, you also set the speed to it's last explicit setting (10 or 100). ■ half – Sets the port to half-duplex mode (port cannot send and receive simultaneously).

Example

```
cli:192.168.208.3:interface:ethernet:csi(1/11/1)# duplex full
cli:192.168.208.3:interface:ethernet:csi(1/11/1)# show interface
Interface                  1/11/1 Ethernet (100 Mb)
Configured duplex                        full
Configured speed                       100 Mb
Auto-negotiation                      disable

In octets                                   0
In unicast                                  0
In multicast                                0
In broadcast                                0

Out octets                                  0
Out unicast                                 0
Out multicast                               0
Out broadcast                               0
```

Related Commands

- **negotiation auto** on page 259
- **show interface ethernet** on page 435
- **speed** on page 582

enable

Enables administrative access to the command line interface. You can only use CLI commands after you log in to the administration console using this command. Upon initial Telnet into the system, you would use this command to access the Cuda12000 command line interface environment.

To access the command line interface, use the **enable** command to log into the CLI environment. Note that you can use the **no enable** command to log out of the CLI environment.

Mode

Any.

Syntax

```
enable <username>
```

You are then prompted for the administrative account password.

```
password <password>
```

Arguments

<username>	User name as assigned by the system administrator.
<password>	Password for the specified user account.

Example

```
cli:null:root> enable root
password: ***
```

Related Commands

- **no enable** on page 269

event-config reporting

Controls how different classes of events are reported, as required by DOCSIS 1.1. For example, you can specify that all events in the alert class are sent to the syslog and to SNMP management stations (in the form of traps).

Mode

`root`

Syntax

```
event-config reporting {default | {{emergency | alert | critical |
  error | warning | notice | info | debug} {none | local |
  local|traps | local|syslog | local|syslog|traps}}
```

Arguments

default Specifies the default reporting configuration:

- emergency – local
- alert – local
- critical – local|syslog|traps
- error – local|syslog|traps
- warning – local|syslog|traps
- notice – local|syslog|traps
- info – none
- debug – none

Event Classes:

emergency Specifies the emergency event class, which is the event class with the highest priority (that is, the most severe events). Events in this class are reserved for fatal hardware or software errors that prevent normal system operation and cause the device to reboot.

alert Specifies the alert event class, which is the event class with the second highest priority. Events in this class indicate a serious failure that causes the device to reboot, but are not caused by hardware or software malfunctioning.

critical	Specifies the critical event class, which is the event class with the third highest priority. Events in this class indicate a serious failure that requires attention and prevents the device from transmitting data, but cannot be recovered without rebooting the system.
error	Specifies the error event class, which is the event class with the fourth highest priority. Events in this class indicate that a failure occurred that could interrupt the normal data flow.
warning	Specifies the warning event class, which is the event class with the fifth highest priority. Events in this class indicate that a failure occurred that could interrupt the normal data flow. These events do not cause a cable modem to re-register.
notice	Specifies the notice event class, which is the event class with the sixth highest priority. Events in this class indicate an event of importance that is not a failure and can be reported in real time through SNMP traps or syslog messages. Examples of events in this class include cold start, warm start, and link up events.
info	Specifies the informational event class, which is the event class with the seventh highest priority. Events in this class are not important and do not indicate failures. However, events in this class may be helpful for tracing normal operations.
debug	Specifies the debug event class, which is the event class with the least priority (that is, the least severe events). Events in this class are non-critical and used for debugging purposes.

Reporting Actions:

none	Events in the specified event class are not reported.
local	All events in the specified event class are written to the internal Cuda 12000 event log.
local|traps	All events in the specified event class are written to the internal Cuda 12000 event log and are sent as SNMP traps. Note that the pipe (|) must be included in the command string.
local|syslog	All events in the specified event class are written to the internal Cuda 12000 event log and are sent as messages to the syslog server. Note that the pipe (|) must be included in the command string.
local|syslog|traps	All events in the specified event class are written to the internal Cuda 12000 event log, are sent as messages to the syslog server, and are sent as SNMP traps. Note that the pipe (|) must be included in the command string.

Example

```
cli:192.168.208.3:root# show event-config reporting
Event Reporting Priorities
--------------------------

row count: 8

Priority    Action
----------- ------------------
  emergency local
      alert local
   critical local|traps|syslog
      error local|traps|syslog
    warning local|traps|syslog
     notice local|traps|syslog
information none
      debug none

cli:192.168.208.3:root# event-config reporting emergency local|syslog|traps
cli:192.168.208.3:root# show event-config reporting
Event Reporting Priorities
--------------------------

row count: 8

Priority    Action
----------- ------------------
  emergency local|traps|syslog
      alert local
   critical local|traps|syslog
      error local|traps|syslog
    warning local|traps|syslog
     notice local|traps|syslog
information none
      debug none
```

Related Commands

- **event-config syslog** on page 162
- **event-config throttle** on page 163
- **show event-config** on page 404

event-config syslog

Specifies the IP address of the system log (syslog) server to which events are sent, as required by DOCSIS 1.1

Mode

```
root
```

Syntax

```
event-config syslog <ip-address>
```

Arguments

<ip-address>	IP address of the syslog server.

Example

```
cli:192.168.208.3:root# event-config syslog 133.132.1.1
cli:192.168.208.3:root#
```

Related Commands

- **event-config reporting** on page 159
- **event-config throttle** on page 163
- **show event-config** on page 404

event-config throttle

Configures the pace of event transmission, as required by DOCSIS 1.1.

Mode

`root`

Syntax

```
event-config throttle {threshold <number> |
  interval <number> | admin {unconstrained | maintainBelowThreshold |
  stopAtThreshold | inhibited}}
```

Arguments

threshold <number>	Specifies the number of events that the Cuda 12000 may generate per event interval before throttling occurs. Throttling is the process of eliminating excessive events. Note that an event causing both a trap and a syslog message is still treated as a single event. Values range from 0 to 4294967295. The default is 0.
interval <number>	Specifies the interval, in seconds, over which the event threshold applies. For example, if you configure an event threshold of 20 and an event interval of 40 seconds, then the Cuda 12000 may generate 20 events over 40 seconds before throttling occurs. Values range from 0 seconds to 2147483647 seconds. The default is 1.
admin {**unconstrained** | **maintainBelowThreshold** | **stopAtThreshold** | **inhibited**}	Controls the transmission of traps and syslog messages with respect to the event threshold. Specify one of these administrative status values: ■ unconstrained (default) – The Cuda 12000 transmits traps and syslog messages without regard to the threshold settings. ■ maintainBelowThreshold – The Cuda 12000 suppresses traps and syslog messages if the number of events exceeds the threshold. ■ stopAtThreshold – The Cuda 12000 stops trap transmissions and syslog messages at the threshold. ■ inhibited – The Cuda 12000 suppresses all trap transmissions and syslog messages.

Example

```
cli:192.168.208.3:root# event-config throttle threshold 5
cli:192.168.208.3:root# event-config throttle interval 100
cli:192.168.208.3:root# event-config throttle admin maintainBelowThreshold
```

Related Commands

- **event-config reporting** on page 159
- **event-config syslog** on page 162
- **show event-config** on page 404

event-log clear

Empties the internal event log.

Mode

`root`

Syntax

`event-log clear`

Example

```
cli:192.168.220.230:root# event-log clear
```

Related Commands

- **show event-log** on page 407

export

Enters export mode. From within this mode you can create export Routing Information Protocol (RIP) or Open Shortest Path First (OSPF) route filters.

Mode

To create OSPF export route filters:

```
router:ospf
```

To create RIP export route filters:

```
router:rip
```

Syntax

```
export
```

Example

```
cli:172.16.19.10:root# router
mode: router
cli:172.16.19.10:router# ospf
mode: router:ospf
cli:172.16.19.10:router:ospf# export
mode: router:ospf:export
```

Related Commands

- **map-list** on page 251
- **match** on page 254
- **route-map** on page 354

filter-aging

Configures IP packet filter aging for all interfaces in the current slot. Specifically this command allows you to enable or disable filter aging, as well as configure the filter aging rate for all interfaces in a specified slot.

You create filters using the **access-list** command and apply them to interfaces using the **access-class** command. When packet filtering is enabled on an inbound or outbound interface, the system compares each packet with all filters applied to the interface. When a match is found, the interface takes the action defined by the filter then adds both the packet and the action to take to a flow table. When the system sees similar packets, it does not have to search through the filters, but instead knows immediately what to do with the packets by referencing the flow table entry.

The filter aging rate is defined as the number of entries aged out of the flow table per second. When an entry is aged out of the flow table, the system can no longer "recognize" similar packets without referencing the filters applied to the interface.

Mode

```
slot (c/s)
```

Syntax

```
filter-aging {in | out} {disable | enable | rate <seconds>}
```

Arguments

in	Specifies all inbound interfaces within the current slot.
out	Specifies all outbound interfaces within the current slot.
disable	Disables filter aging for the specified interfaces.
enable	Enables filter aging for the specified interfaces.
rate <seconds>	Filter aging rate in seconds. Sets the number of seconds before a flow (flow table entry) is aged out (removed).

Example

The following example enables filter aging for all outbound interfaces in slot 1/1/1 then sets the rate at which outbound traffic flows are aged out of the flow table to 500 seconds.

```
cli:172.16.19.10:slot(1/1)# filter-aging out enable
cli:172.16.19.10:slot(1/1)# filter-aging out rate 500
cli:172.16.19.10:slot(1/1)# show filter-aging

IP Filter Aging In                   disable
IP Filter Aging In Rate                 4096
IP Filter Aging Out                   enable
IP Filter Aging Out Rate                 500

cli:172.16.19.10:slot(1/1)#
```

Related Commands

- **access-class** on page 45
- **access-list** on page 47
- **show filter-aging** on page 409
- **slot** on page 555

flap-list aging

Specifies the number of days to retain flapping information on cable modems currently in the flap list table.

The system maintains a cable modem flap table for every cable modem (whether active or not) that has difficulty communicating with the CMTS. Flapping refers to the rapid disconnecting and reconnecting of cable modems. The flap list contains modem MAC addresses and logs the time of the most recent flapping activity on a per-modem basis.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
flap-list aging <value>
```

Arguments

<value>	Number of days that you want the system to maintain flapping information. The system ages out any flapping information that has been stored in the table beyond this time frame. Valid values: 1 – 60.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# flap-list aging 20
```

Related Commands

- **flap-list clear** on page 170
- **flap-list insertion-time** on page 172
- **flap-list power-adj-threshold** on page 174
- **flap-list size** on page 175
- **show flap-list** on page 410
- **show interface cable <c/s/i> flap-list** on page 420

flap-list clear

Clears the flap list of all cable modems on a the current CMTS card. Clearing the flap list with this command sets the flap count back to zero for all modems attached to the current CMTS.

The system maintains a cable modem flap table for every cable modem (whether active or not) that has difficulty communicating with the CMTS. Flapping refers to the rapid disconnecting and reconnecting of cable modems. The flap list maintains a flap count and logs the most recent flapping activity for each cable modem attached to the CMTS.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

`flap-list clear`

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# show flap-list

row count: 14

Mac Address       Flap  Last Known    Insert Time    Remove Time    Hit   Miss
                  Count State                                     Count Count
----------------- ----- ------------- -------------- -------------- ----- -----
00:10:95:01:ef:d8     3 Registered    99-08-24 16:39 99-08-24 16:53   694    32
00:10:95:01:f0:05     4 Registered    99-08-24 16:39 99-08-24 16:53   958    32
00:10:95:04:0a:b7     3 Registered    99-08-24 16:39 99-08-24 16:53   694    32
00:10:95:04:0a:bd     3 Registered    99-08-24 16:39 99-08-24 16:53   694    46
00:10:95:04:0a:c3     3 Registered    99-08-24 16:39 99-08-24 16:53   694    32
00:10:95:04:0a:c4     3 Registered    99-08-24 16:39 99-08-24 16:53   695    32
00:90:83:32:9f:8c     3 Registered    99-08-24 16:40 99-08-24 16:53   697    33
00:90:83:36:82:ee     3 Registered    99-08-24 16:39 99-08-24 16:53   701    44
00:90:83:36:82:f1     3 Registered    99-08-24 16:39 99-08-24 16:53   702    47
00:90:96:00:29:6d     5 RegBpiTek     99-08-24 16:38 99-08-24 16:53  1199    48
00:90:96:00:29:71    16 RegBpiTek     99-08-24 16:39 99-08-24 16:53   716    98
00:90:96:00:39:7f     9 Registered    99-08-24 16:39 99-08-24 16:53  1207    65
00:90:96:00:39:f9    16 Registered    99-08-24 16:39 99-08-24 16:53   715   102
00:a0:73:69:39:65     4 RegBpiTek     99-08-24 16:40 99-08-24 16:53   694    32

cli:172.16.19.10:interface:cable:csi(1/1/1)# flap-list clear
cli:172.16.19.10:interface:cable:csi(1/1/1)# show flap-list


Mac Address       Flap  Last Known    Insert Time    Remove Time    Hit   Miss
                  Count State                                     Count Count
----------------- ----- ------------- -------------- -------------- ----- -----

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **flap-list aging** on page 169
- **flap-list insertion-time** on page 172
- **flap-list power-adj-threshold** on page 174
- **flap-list size** on page 175
- **show flap-list** on page 410
- **show interface cable <c/s/i> flap-list** on page 420

flap-list insertion-time

This command sets the flap-list insertion time, in seconds, for the current CMTS.

The system maintains a cable modem flap table for every cable modem (whether active or not) that has difficulty communicating with the CMTS. Flapping refers to the rapid disconnecting and reconnecting of cable modems. The flap list maintains a flap count and logs the most recent flapping activity for each cable modem attached to the CMTS.

The insertion time is the primary criteria used in determining which cable modems the system adds to the flap list. The system adds a modem to the flap list if the modem comes online, and then goes offline within the amount of time specified by this command.

For example, if the flap list insertion time is set to 60 seconds, and a modem goes offline within 45 seconds of coming online, the system records the flap event in the flap list. If instead the modem goes offline 65 seconds after coming online, the flap event is not recorded.

Note that a higher flap list insertion time results in the system recording a greater number of flap events; a lower number results in the system recording fewer flap events.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
flap-list insertion-time <value>
```

Arguments

<value>	Flap insertion time, in seconds, that you want to specify for this CMTS interface. Default: 604800 seconds. Acceptable values: 0 – 604800.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# flap-list insertion-time 604800
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **filter-aging** on page 167
- **flap-list clear** on page 170
- **flap-list power-adj-threshold** on page 174
- **flap-list size** on page 175
- **show flap-list** on page 410
- **show interface cable <c/s/i> flap-list** on page 420

flap-list power-adj-threshold

Sets the cable flap list power adjustment threshold.

The system maintains a cable modem flap table for every cable modem (whether active or not) that has difficulty communicating with the CMTS. Flapping refers to the rapid disconnecting and reconnecting of cable modems. The flap list maintains a flap count and logs the most recent flapping activity for each cable modem attached to the CMTS.

If a modem is already in the flap list, and requires a ranging adjustment greater than the power-adj-threshold, then that modem's power adjust count is incriminated in the flap list.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
flap-list power-adj-threshold <value>
```

Arguments

<value>	Enter the number of decibels to which you want to set the cable modem flap list. Valid range: 1 – 10 dbmv.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# flap-list power-adj-threshold 9
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **flap-list aging** on page 169
- **flap-list clear** on page 170
- **flap-list insertion-time** on page 172
- **flap-list size** on page 175
- **show flap-list** on page 410
- **show interface cable <c/s/i> flap-list** on page 420

flap-list size

Sets the maximum number entries in the flap list table on the current CMTS card.

The system maintains a cable modem flap table for every cable modem (whether active or not) that has difficulty communicating with the CMTS. Flapping refers to the rapid disconnecting and reconnecting of cable modems. The flap list maintains a flap count and logs the most recent flapping activity for each cable modem attached to the CMTS.

When the number of cable modems recorded in the flap list exceeds the threshold that you specify with this command, no more new flap entries will be added to the flap table.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
flap-list size <value>
```

Arguments

<value>	Maximum number of entries that the flap list can hold before aging out the oldest entries. Valid range: 0 – 8192. Default: 8191.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# flap-list size 300
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **flap-list aging** on page 169
- **flap-list clear** on page 170
- **flap-list insertion-time** on page 172
- **flap-list power-adj-threshold** on page 174
- **show flap-list** on page 410
- **show interface cable <c/s/i> flap-list** on page 420

help

Displays commands available in the current mode and provides a brief description of each command.

Mode

Any.

Syntax

help [<command>]

Arguments

<command>	Specifies a command for which you want help.

Example

```
cli:192.168.208.3:root# router
mode: router
cli:192.168.208.3:router# help
  basmonitor                  Start the monitor


  boot                        Allows a module in an application slot to be
                              enabled and disabled
  clear
  connect                     Connect to remote system
  echo                        Echo a comment to display
  enable                      Logon to enable session
  interface                   Change to interface mode
  ip
  no                          Deletion
  ospf                        Enter router-OSPF mode
  ping                        For Chassis Controller to be able to perform
                              ping from CC
  prov-server                 Change to provision server mode
  q                           Exit from cli shell
  quit                        Exit from cli shell
  rip                         Enter router-RIP mode
  root                        Change to root mode
  router                      Change to router mode
  server                      Change to server mode
  set                         Set parameter
  show                        Display Information
  sleep                       Wait for specified number of seconds
  slot                        Change to slot mode
  source                      Execute a script file
  traceroute                  traceroute [-...] <ip_address> [size]
  up                          Change mode to one level up
cli:192.168.208.3:router#
```

http-server

Enables and disables the HTTP server on the Cuda 12000.

Mode

root

Syntax

http-server {**enable** | **disable**}

Arguments

enable	Enables the HTTP server.
disable	Disables the HTTP server. If you disable the HTTP server, you cannot use a Web browser to manage the Cuda 12000 IP Access Switch.

Example

```
cli:192.168.208.3:root# http-server disable
Stopping http server ...
Shutting down http: [  OK  ]
cli:192.168.208.3:root# http-server enable
Starting http server ...
Starting httpd: [  OK  ]
```

import

Enters import mode. From within this mode you can create Routing Information Protocol (RIP) or Open Shortest Path First (OSPF) import route filters.

Mode

To create OSPF import route filters/templates:

```
router:ospf
```

To create RIP import route filters/templates:

```
router:rip
```

Syntax

```
import
```

Example

```
cli:172.16.19.10:root# router
mode: router
cli:172.16.19.10:router# ospf
mode: router:ospf
cli:172.16.19.10:router:ospf# import
mode: router:ospf:import
```

Related Commands

- **map-list** on page 251
- **match** on page 254
- **route-map** on page 354

insertion-interval

Sets the modem insertion interval for the current DOCSIS module. When a cable begins ranging, it sends an initial Ranging Request within an Initial Maintenance Region. The insertion interval controls how frequently Initial Maintenance Regions are scheduled by the CMTS.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

insertion-interval <centiseconds>

Arguments

<centiseconds>	Cable insertion interval in centiseconds. This defines a limit to the amount of time a cable modem can initially request a channel from the CMTS. Valid range: 1 – 200 centiseconds.

Example

```
cli:172.16.19.10:router:rip# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# insertion-interval 50
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421
- **show mac** on page 454

interface

Use this command to enter interface configuration mode for a specific interface:

- Ethernet
- Cable
- POS
- Bridge-Group
- Loopback

Mode

Any.

Syntax

```
interface {<chassis>/<slot>/<interface> | loopback}
```

Arguments

{<chassis>/<slot>/<interface> | **loopback**}	Specifies one of the following values: ■ <chassis>/<slot>/<interface> – Specifies the combination of chassis number, slot number, and interface number (port number) that identifies the Ethernet, cable, POS, or bridge-group interface you want to configure. The chassis number can range from 1 to 128, though currently it can only be 1. The slot number can range from 1 to 12. The range of interface numbers depends on the number of ports on the card. ■ loopback – Specifies the loopback interface, to enter loopback mode. You can then manage this interface as you would any other IP interface. Note that this loopback interface differs from the standard IP loopback interface (127.0.0.1). One of the uses of this loopback interface is for opening in-band connections (such as telnet sessions) to the Cuda 12000. Users can specify the IP address of the loopback interface to open connections, thereby eliminating the need to specify the IP address of a physical interface (such as a CMTS interface).

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **show interface** on page 414
- **show interface cable <c/s/i> cm-filter-default** on page 415
- **show interface cable <c/s/i> cpe-control** on page 416
- **show interface cable <c/s/i> downstream** on page 417
- **show interface cable <c/s/i> dynamic-service-stats** on page 418
- **show interface cable <c/s/i> flap-list** on page 420
- **show interface cable <c/s/i> mac** on page 421
- **show interface cable <c/s/i> modem** on page 423
- **show interface cable <c/s/i> modem summary** on page 424
- **show interface cable <c/s/i> modulation-profile** on page 425
- **show interface cable <c/s/i> pll-state** on page 426
- **show interface cable <c/s/i> proxy-arp** on page 427
- **show interface cable <c/s/i> qos** on page 428
- **show interface cable <c/s/i> sid** on page 429
- **show interface cable <c/s/i> signal-quality** on page 431
- **show interface cable <c/s/i> upstream** on page 433
- **show interface ethernet** on page 435
- **show interface pos** on page 437

ip address

Adds an IP address to the current interface and enters configuration mode for that IP interface.

Mode

```
interface:<type>:csi(<c/s/i>)
```

Syntax

```
ip address <ip-address> <mask> [{other | secondary | primary}]
```

Arguments

<ip-address>	IP address that you want to add to the current interface.
<mask>	Network mask for the specified interface that you want to add to the interface.
{**other** | **primary** | **secondary**}	Specifies one of the following values: ■ other – Specify this parameter if you are adding the IP address of a network other than the primary or secondary on the current interface. ■ primary – Specify this parameter if you are adding the IP address of a primary network on the current interface. If you do not specify an optional parameter (other, secondary, primary), the interface is added as a primary interface by default. Keep in mind that the last IP interface that you add is the primary (by default) unless you specify otherwise. ■ secondary – Specify this parameter if you are adding the IP address of a secondary network on the current interface.

Example

```
cli:192.168.208.3:interface:ethernet:csi(1/11/1)# ip address 192.168.16.3
255.255.255.0
cli:192.168.208.3:interface:ethernet:csi(1/11/1):ip-address(192.168.16.3)#
show ip address
Chassis/Slot/Interface      1/11/1

row count: 1

IP Address       Net Mask         Interface  Priority
---------------- ---------------- ---------- ----------
192.168.16.3     255.255.255.0      11337729    Primary
```

Related Commands

- **no ip address** on page 270
- **show ip address** on page 443

ip filter

Enables and disables IP packet filtering on the selected cable interface. Access lists can be applied to incoming or outgoing packets. When disabled, any access lists applied to an interface are ignored and all packets pass. If enabled, the filtering rules are applied.

Note that filtering is automatically enabled when you apply a filter to an interface using the **access-class** command, but it is not automatically disabled when the filter is removed. To disable packet filtering, you must do so with the **ip filter** command.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
ip filter {in | out} {enable | disable}
```

Arguments

in	Specified the inbound interface.
out	Specified the outbound interface
enable	Enables packet filtering on the inbound or outbound interface.
disable	Disables packet filtering on the inbound or outbound interface.

Example

The following example enables packet filtering of inbound packets on cable interface 1/1/1 and disables filtering of outbound packets on the same interface.

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# ip filter in enable
cli:172.16.19.10:interface:cable:csi(1/1/1)# ip filter out disable
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **access-class** on page 45
- **access-list** on page 47
- **show access-class** on page 369
- **show access-list** on page 370
- **show ip filter** on page 444

ip igmp

Configures the interface for IGMP communications.

Mode

`interface`

Syntax

```
ip igmp {join-group <group-address> | query-interval <seconds> |
  query-max-response-time <seconds> | version {2 | 1 | v2_only} |
  robustness <value> | router | last-query-interval <seconds>}
```

Arguments

join-group <group-address>	Specifies the Class D IP address of the multicast group (for example, 225.3.2.2) that the Cuda 12000 joins on the interface.
query-interval <seconds>	Specifies the frequency, in seconds, that the Cuda 12000 transmits IGMP host query packets on this particular interface. The default is 125 seconds with a range of 10 to 65535 seconds.
query-max-response-time <seconds>	Specifies the maximum number of seconds that the Cuda 12000 waits for a response to an IGMP Query message before deleting the group. The default is 10 seconds with a range of 1 to 25 seconds.
version {**2** | **1** | **v2_only**}	Version of IGMP running on this particular interface. For IGMP to function properly, all routers on a network must be configured to run the same version of IGMP. The default is 2 and the possible values are: ■ 2 – Version 2. If the Cuda 12000 encounters another host or router on the network using Version 1, the Cuda 12000 reverts back to using Version 1. ■ 1 – Version 1. ■ v2_only – Version 2 only. If the Cuda 12000 encounters another host or router on the network using Version 1, the Cuda 12000 continues to run Version 2.
robustness <number>	Allows you to compensate for the expected packet loss on a subnet. Increase the value if you expect the loss to be high. The default is 2 and the range is 1 to 255.

router	Enables the interface to function as an IGMP Querier (router). For the DOCSIS module, the default is router. Use the **no igmp router** command to disable the IGMP Querier capability on the interface. This causes the interface to function as an IGMP host. If multiple routers attempt to become the IGMP Querier, the one with the lowest IP address becomes the querier. When you change an interface from a querier to a host, any multicast groups that are learned are removed. When you change the interface from a host to a querier, or from a querier to a host, any multicast groups joined locally or by an application running on the Cuda 12000 remain. **Note**: The DOCSIS module can only be configured as an IGMP querier (router). All other interfaces default to IGMP hosts.
last-query-interval <seconds> **(Does not apply to IGMP Version 1)**	Specifies the number of seconds between queries to find other hosts on the network that want to receive traffic from the multicast group. You can tune this parameter to modify the leave latency of the network. A reduced value results in reduced time to detect the loss of the last member of a group. The default is 1 and the range is 1 to 25 seconds.

Example 1

In this example, the multicast group 225.2.2.1 is joined on interface 1/3/1:

```
cli:192.168.208.3:interface:ethernet:csi(1/3/1)# ip igmp join-group 225.2.2.1
cli:192.168.208.3:interface:ethernet:csi(1/3/1)#
```

Example 2

In this example, the user configures interface 1/3/1 for IGMP Version 2 only:

```
cli:192.168.208.3:interface:ethernet:csi(1/3/1)# ip igmp version v2_only
cli:192.168.208.3:interface:ethernet:csi(1/3/1)#
```

Related Commands

- **no ip igmp** on page 271
- **show ip {mroute | igmp}** on page 439

ip igmp proxy

Configures interfaces to act as IGMP proxies for a single multicast group or a range of multicast groups.

Mode

`root`

Syntax

`ip igmp proxy` <group-address> <group-mask> **`metric`** <number> <c/s/i>

Arguments

<group-address>	Class D IP address (for example, 239.1.1.1) of the multicast group for which the interface acts as a proxy. For proxy addresses, note that you cannot use an address in the well-known multicast address range 224.0.0.0 to 224.0.0.255.
<group-mask>	A mask, in dot-notation format, that can be used with a single multicast group address or to specify a multicast address range. For example, a multicast address of 239.1.0.0 and a mask of 255.255.0.0 means that the interface proxies for all multicast groups from 239.1.0.0 to 239.1.255.255. For a single multicast group address (for example, 239.2.2.2), specify a mask of 255.255.255.255.
metric <number>	Metric value from 1 to 255 that assigns a priority to the proxy. One (1) is the highest priority; 255 is the lowest priority.
<c/s/i>	The physical interface that acts as an IGMP proxy (for example, 1/2/4).

Example 1

In this example, the user configures interface 1/1/1 to act as an IGMP proxy for the multicast group 225.4.3.4:

```
cli:root# ip igmp proxy 225.4.3.4 255.255.255.255 metric 3 1/1/1
cli:root#
```

Example 2

In this example, the user configures two interfaces to act as proxies. Each interface acts as a proxy for a range of multicast groups. One interface acts as a proxy for all multicast groups in the range 225.4.3.0 to 225.4.3.255. The other interface acts as a proxy for all multicast groups in the range 225.4.0.0 to 225.4.255.255:

```
cli:192.168.208.3:root# ip igmp proxy 225.4.3.0 255.255.255.0 metric 1 1/1
cli:192.168.208.3:root# ip igmp proxy 225.4.0.0 255.255.0.0 metric 1 1/2/1
```

Notice that the two ranges overlap, and that the range 225.4.3.0 to 225.4.3.255 is more specific than the other range. By taking this approach, the user guarantees that a more specific route takes precedence over a less specific route. For example, the Cuda 12000 proxies multicast packets addressed to 225.4.3.1 on interface 1/1/1; however, the Cuda 12000 would proxy a multicast packet addressed to 225.4.5.1 on interface 1/2/1.

Related Commands

- **no ip igmp proxy** on page 273
- **show ip {mroute | igmp}** on page 439

ip ospf area-id

Use this command to configure the OSPF area ID on the current interface. Note that you must have already created the area ID that you want to assign to the current interface with the **ospf area** command within router:ospf mode.

You execute this command from IP interface mode. Once within this mode, you must specify the IP interface to which you want to apply the OSPF configuration using the **ip address** command. You can verify the IP address that you are configuring using the **show mode** command, which includes an IP Address=<ip address> line. This line indicates the IP interface that you are currently configuring. If you fail to specify the IP interface and enter IP interface configuration mode, the system will not react this **ip ospf** command.

Use the **no ip ospf area-id** command to remove an area for the current interface.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

ip ospf area-id <area-id> [{**enable** | **disable**}]

no ip ospf area-id <area-id>

Arguments

<area-id>	Area ID to which you want this interface to belong.
enable	Enables the area ID.
disable	Disables the area ID.

Example

The following example enters router ospf mode to create the area ID 1.1.1.3, then assigns that area ID to IP interface 172.16.19.89 on physical interface 1/11/4:

```
cli:root# router ospf
mode: router:ospf
cli:router:ospf# ospf area 1.1.1.3
cli:router:ospf# interface 1/11/4
mode: interface:ethernet:csi(1/11/4)
cli:interface:ethernet:csi(1/11/4)# ip address 172.16.19.89 255.255.255.0
cli:interface:ethernet:csi(1/11/4):ip-address(172.16.19.89)# ip ospf area-id
1.1.1.3
cli:interface:ethernet:csi(1/11/4):ip-address(172.16.19.89)#
```

Related Commands

- **ip ospf authentication** on page 193
- **ip ospf cost** on page 196
- **ip ospf dead-interval** on page 199
- **ip ospf hello-interval** on page 202
- **ip ospf priority** on page 205
- **ip ospf retransmit-interval** on page 208
- **ip ospf transit-delay** on page 211
- **router** on page 356
- **show ip ospf** on page 448

ip ospf authentication

Specifies the type of authentication (MD5 or simple password) that OSPF uses as a security measure so that the current interface exchanges routing information with authorized neighbors only. This command also specifies the security key that is used for routing information exchanges. Make sure that:

- The Cuda 12000 and all of its OSPF neighbors on the current interface use the same authentication type, key ID (for MD5 only), and key.
- The authentication type for the area ID associated with the current interface matches the authentication type you configure for the interface. For example, if you configure MD5 authentication for an area ID, 1.1.1.1, and you associate area ID 1.1.1.1 with the current interface, then you must configure MD5 authentication for the current interface. Refer to **ospf area** on page 307 for more information on configuring authentication types for areas.

Use the no form of the command to disable authentication on an interface.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

```
ip ospf authentication {md5 | password | key-id <id> key <key> |
  key <key>}

no ip ospf authentication
```

Arguments

md5

Specifies that the OSPF interface uses MD5 authentication. MD5 authentication employs a shared secret key to generate a message digest, which is a 128-bit checksum of the packet and key. When a router sends a packet, it includes:

- The message digest
- A key ID that identifies the key that generated the message digest

At first, the receiving router accepts any packet that contains a key ID that matches one of its own key IDs. Using this key ID, the receiving router looks up the key. The receiving router uses the key associated with the received key ID to replace the message digest in the packet and generates a new message digest. If the generated message digest matches the message digest in the received packet, then the router accepts the packet. Otherwise, the router rejects the packet.

For more information on MD5 authentication, see RFC 1321 (The MD5 Message-Digest Algorithm) and RFC 2328 (OSPF Version 2), which provides information on how MD5 authentication is used with OSPF.

If you specify md5 as the authentication type, you must subsequently specify the shared secret key with the key-id <id> key <key> argument.

password

Specifies a simple password authentication method that includes a password in all OSPF messages. When an interface receives a message that uses simple password authentication, the Cuda 12000 checks the incoming OSPF message to see if the password is included. If the password is correct, the Cuda 12000 processes the message normally. If the password is not part of the incoming message, the Cuda 12000 drops the message.

If you specify password as the authentication type, you must subsequently specify the password with the key <key> argument.

key-id <id> **key** <key>

Specifies the md5 key ID and shared secret key:

- key-id <id> – An integer ranging from 1 to 255. This value identifies the shared secret key.
- key <key> – Specifies the key, which is a text string from 1 to 16 characters.

key <key>

Specifies the simple password, which is a text string from 1 to 16 characters.

Example 1

In this example, the user configures MD5 authentication:

```
cli:# ip ospf authentication md5
cli:# ip ospf authentication key-id 45 key my-secret-key
cli:#
```

Example 2

In this example, the user configures simple password authentication:

```
cli:# ip ospf authentication password
cli:# ip ospf authentication key secretky
cli:#
```

Related Commands

- **ip ospf area-id** on page 191
- **ip ospf cost** on page 196
- **ip ospf dead-interval** on page 199
- **ip ospf hello-interval** on page 202
- **ip ospf priority** on page 205
- **ip ospf retransmit-interval** on page 208
- **ip ospf transit-delay** on page 211
- **router ospf** on page 359
- **show ip ospf** on page 448

ip ospf cost

Configures the cost metric for the current OSPF interface. Note the following:

- You execute this command from IP interface mode.
- Once within this mode, you must specify the IP interface to which you want to apply the OSPF configuration using the **ip address** command.
- You must have already defined one or more ospf areas with the **ospf area** command within router:ospf mode. You must apply an ospf area to the current interface using the **ip ospf area-id** before you use this command.
- Use the **show ip ospf** command to display the OSPF configuration for the current interface.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

```
ip ospf cost <cost>
```

Arguments

<cost>	Cost that you want to assign to this OSPF interface. Valid range is 0 to 65535. The default is 1. This value specifies the type of service (TOS) cost. The lowest TOS cost has the highest priority for routing. For example, if two interfaces can be used to forward traffic to the same destination, the one with the lower TOS cost is selected.

Example

```
cli:# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli:# show ip ospf
IP Address                          201.1.1.1
Area ID                               0.0.0.1
Type                                    Bcast
Priority                                    5
Transit Delay                               1
Retrans Int                                 5
Hello Int                                  10
Dead Int                                   40
Poll Int                                    0
Admin Stat                            Enabled
Status                                 Active
Auth Type                                None

Auth Key Id                                 0
Cost                                        1
cli:# ip ospf cost 2
cli:# show ip ospf
IP Address                          201.1.1.1
Area ID                               0.0.0.1
Type                                    Bcast
Priority                                    5
Transit Delay                               1
Retrans Int                                 5
Hello Int                                  10
Dead Int                                   40
Poll Int                                    0
Admin Stat                            Enabled
Status                                 Active
Auth Type                                None

Auth Key Id                                 0
Cost                                        2
cli:#
```

Related Commands

- **ip ospf area-id** on page 191
- **ip ospf authentication** on page 193
- **ip ospf dead-interval** on page 199
- **ip ospf hello-interval** on page 202
- **ip ospf priority** on page 205
- **ip ospf retransmit-interval** on page 208
- **ip ospf transit-delay** on page 211
- **router ospf** on page 359
- **show ip ospf** on page 448

ip ospf dead-interval

Configures the dead-interval for the current OSPF interface. The dead interval is the interval at which hello packets must not be seen before neighbors declare the router down. Note the following:

- You execute this command from IP interface mode.
- Once within this mode, you must specify the IP interface to which you want to apply the OSPF configuration using the **ip address** command.
- You must have already defined one or more ospf areas with the **ospf area** command within router:ospf mode. You must apply an ospf area to the current interface using the **ip ospf area-id** before you use this command.
- Use the **show ip ospf** command to display the OSPF configuration for the current interface.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

```
ip ospf dead-interval <seconds>
```

Arguments

<seconds>	Interval, in seconds. You must set the same value for all participating OSPF routers on the network. Valid range is 0 to 2147483647 seconds. The default is 40.

Example

```
cli:# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli:# show ip ospf
IP Address                          201.1.1.1
Area ID                               0.0.0.1
Type                                    Bcast
Priority                                    5
Transit Delay                               1
Retrans Int                                 5
Hello Int                                  10
Dead Int                                   40
Poll Int                                    0
Admin Stat                            Enabled
Status                                 Active
Auth Type                                None

Auth Key Id                                 0
Cost                                        2
cli:# ip ospf dead-interval 50
cli:# show ip ospf
IP Address                          201.1.1.1
Area ID                               0.0.0.1
Type                                    Bcast
Priority                                    5
Transit Delay                               1
Retrans Int                                 5
Hello Int                                  10
Dead Int                                   50
Poll Int                                    0
Admin Stat                            Enabled
Status                                 Active
Auth Type                                None

Auth Key Id                                 0
Cost                                        1
```

Related Commands

- **ip ospf area-id** on page 191
- **ip ospf authentication** on page 193
- **ip ospf cost** on page 196
- **ip ospf hello-interval** on page 202
- **ip ospf priority** on page 205
- **ip ospf retransmit-interval** on page 208
- **ip ospf transit-delay** on page 211
- **router ospf** on page 359
- **show ip ospf** on page 448

ip ospf hello-interval

Specifies the length of time, in seconds, the router waits between sending Hello packets on the current OSPF interface. Note the following:

- You execute this command from IP interface mode.
- Once within this mode, you must specify the IP interface to which you want to apply the OSPF configuration using the **ip address** command.
- You must have already defined one or more ospf areas with the **ospf area** command within router:ospf mode. You must apply an ospf area to the current interface using the **ip ospf area-id** command before you use this command.
- Use the **show ip ospf** command to display the OSPF configuration for the current interface.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

```
ip ospf hello-interval <seconds>
```

Arguments

<seconds>	Interval, in seconds, between Hello packet transmissions. Valid range is 1 to 65535 seconds. The default is 10.

Example

```
cli:# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli:# show ip ospf
IP Address                          201.1.1.1
Area ID                               0.0.0.1
Type                                    Bcast
Priority                                    5
Transit Delay                               1
Retrans Int                                 5
Hello Int                                  10
Dead Int                                   50
Poll Int                                    0
Admin Stat                            Enabled
Status                                 Active
Auth Type                                None

Auth Key Id                                 0
Cost                                        2
cli:# ip ospf hello-interval 15
cli# show ip ospf
IP Address                          201.1.1.1
Area ID                               0.0.0.1
Type                                    Bcast
Priority                                    5
Transit Delay                               1
Retrans Int                                 5
Hello Int                                  15
Dead Int                                   50
Poll Int                                    0
Admin Stat                            Enabled
Status                                 Active
Auth Type                                None

Auth Key Id                                 0
Cost                                        2
```

Related Commands

- **ip ospf area-id** on page 191
- **ip ospf authentication** on page 193
- **ip ospf cost** on page 196
- **ip ospf dead-interval** on page 199
- **ip ospf priority** on page 205
- **ip ospf retransmit-interval** on page 208
- **ip ospf transit-delay** on page 211
- **router ospf** on page 359
- **show ip ospf** on page 448

ip ospf priority

Assigns the interface priority to the current OSPF interface. The interface priority determines which OSPF router is elected as the designated router. Note the following:

- You execute this command from IP interface mode.
- Once within this mode, you must specify the IP interface to which you want to apply the OSPF configuration using the **ip address** command.
- You must have already defined one or more ospf areas with the **ospf area** command within router:ospf mode. You must apply an ospf area to the current interface using the **ip ospf area-id** before you use this command.
- Use the **show ip ospf** command to display the OSPF configuration for the current interface.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

```
ip ospf priority <number>
```

Arguments

<number>	The priority that you want to assign to this OSPF interface. Valid range is 0 to 255. The default is 5. This number identifies the priority of the Cuda 12000 relative to other OSPF routers on the current interface. The number is used to elect the designated and backup designated routers. The router with the highest priority is considered the designated router. A value of 0 indicates that the router is not eligible to be the designated or backup designated router. If all routers have the same priority, the router ID is used to determine the designated router.

Example

```
cli:# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli:# show ip ospf
IP Address                          201.1.1.1
Area ID                               0.0.0.1
Type                                    Bcast
Priority                                    5
Transit Delay                               1
Retrans Int                                 5
Hello Int                                  15
Dead Int                                   50
Poll Int                                    0
Admin Stat                            Enabled
Status                                 Active
Auth Type                                None

Auth Key Id                                 0
Cost                                        2
cli:# ip ospf priority 10
cli:# show ip ospf
IP Address                          201.1.1.1
Area ID                               0.0.0.1
Type                                    Bcast
Priority                                   10
Transit Delay                               1
Retrans Int                                 5
Hello Int                                  15
Dead Int                                   50
Poll Int                                    0
Admin Stat                            Enabled
Status                                 Active
Auth Type                                None

Auth Key Id                                 0
Cost                                        2
```

Related Commands

- **ip ospf area-id** on page 191
- **ip ospf authentication** on page 193
- **ip ospf cost** on page 196
- **ip ospf dead-interval** on page 199
- **ip ospf hello-interval** on page 202
- **ip ospf retransmit-interval** on page 208
- **ip ospf transit-delay** on page 211
- **router ospf** on page 359
- **show ip ospf** on page 448

ip ospf retransmit-interval

Sets the Link State Advertisement (LSA) retransmit interval in seconds for the current interface. This interval determines the number of seconds before a link state advertisement is retransmitted. Note the following:

- You execute this command from IP interface mode.
- Once within this mode, you must specify the IP interface to which you want to apply the OSPF configuration using the **ip address** command.
- You must have already defined one or more ospf areas with the **ospf area** command within router:ospf mode. You must apply an ospf area to the current interface using the **ip ospf area-id** before you use this command.
- Use the **show ip ospf** command to display the OSPF configuration for the current interface.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

```
ip ospf retransmit-interval <seconds>
```

Arguments

<seconds>	The retransmit interval that you want to define for this OSPF interface. Valid range is 1 to 65535. The default is 5.

Example

```
cli:# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli# show ip ospf
IP Address                          201.1.1.1
Area ID                               0.0.0.1
Type                                    Bcast
Priority                                   10
Transit Delay                               1
Retrans Int                                 5
Hello Int                                  15
Dead Int                                   50
Poll Int                                    0
Admin Stat                            Enabled
Status                                 Active
Auth Type                                None

Auth Key Id                                 0
Cost                                        2
cli:# ip ospf retransmit-interval 10
cli# show ip ospf
IP Address                          201.1.1.1
Area ID                               0.0.0.1
Type                                    Bcast
Priority                                   10
Transit Delay                               1
Retrans Int                                10
Hello Int                                  15
Dead Int                                   50
Poll Int                                    0
Admin Stat                            Enabled
Status                                 Active
Auth Type                                None

Auth Key Id                                 0
Cost                                        2
```

Related Commands

- **ip ospf area-id** on page 191
- **ip ospf authentication** on page 193
- **ip ospf cost** on page 196
- **ip ospf dead-interval** on page 199
- **ip ospf hello-interval** on page 202
- **ip ospf priority** on page 205
- **ip ospf transit-delay** on page 211
- **router ospf** on page 359
- **show ip ospf** on page 448

ip ospf transit-delay

Sets the number of seconds it takes to transmit a link state update packet on the current OSPF interface. Note the following:

- You execute this command from IP interface mode.
- Once within this mode, you must specify the IP interface to which you want to apply the OSPF configuration using the **ip address** command.
- You must have already defined one or more ospf areas with the **ospf area** command within router:ospf mode. You must apply an ospf area to the current interface using the **ip ospf area-id** before you use this command.
- Use the **show ip ospf** command to display the OSPF configuration for the current interface.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

```
ip ospf transit-delay <number>
```

Arguments

<number>	Time, in seconds, that it takes to transmit a link state update over this OSPF interface. Valid range is 1 to 65535. The default is 1.

Example

```
cli:# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli:# show ip ospf
IP Address                          201.1.1.1
Area ID                               0.0.0.1
Type                                    Bcast
Priority                                   10
Transit Delay                               1
Retrans Int                                10
Hello Int                                  15
Dead Int                                   50
Poll Int                                    0
Admin Stat                            Enabled
Status                                 Active
Auth Type                                None

Auth Key Id                                 0
Cost                                        2
cli:# ip ospf transit-delay 2
cli:# show ip ospf
IP Address                          201.1.1.1
Area ID                               0.0.0.1
Type                                    Bcast
Priority                                   10
Transit Delay                               2
Retrans Int                                10
Hello Int                                  15
Dead Int                                   50
Poll Int                                    0
Admin Stat                            Enabled
Status                                 Active
Auth Type                                None

Auth Key Id                                 0
Cost                                        2
```

Related Commands

- **ip ospf area-id** on page 191
- **ip ospf authentication** on page 193
- **ip ospf cost** on page 196
- **ip ospf dead-interval** on page 199
- **ip ospf hello-interval** on page 202
- **ip ospf priority** on page 205
- **ip ospf retransmit-interval** on page 208
- **router ospf** on page 359
- **show ip ospf** on page 448

ip rip accept default-route

Configures the interface to accept default routes that neighbor routers advertise. The default route is the one that routers select when they do not have a specific route to a destination network, subnetwork, or host. Use the no form of the command to disable default route acceptance.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

ip rip accept default-route

no ip rip accept default-route

Example

```
cli:interface:ethernet:csi(1/11/1)# ip address 192.168.16.3 255.255.255.0
cli:interface:ethernet:csi(1/11/1):ip-address(192.168.16.3)# ip rip accept
default-route
cli:interface:ethernet:csi(1/11/1):ip-address(192.168.16.3)# show ip rip
IP Address                      192.168.16.3
Send Version                           RIPv2
Receive Version                RIPv1 Or RIPv2
Cost                                       1
Authentication ON                      False
Authentication Type         No Authentication
Authentication Key Id                      0
Send Default Only                      False
Send Default Also                      False
Default Cost                               0
Accept Default                          True
Accept Host Route                       True
Split Horizon                           True
Poisoned Reverse                        True
Status                                Active
```

Related Commands

- **ip rip accept host-route** on page 216
- **ip rip authentication** on page 218
- **ip rip cost** on page 221
- **ip rip default cost** on page 223
- **ip rip disable** on page 225
- **ip rip enable** on page 227
- **ip rip neighbor** on page 229
- **ip rip poisoned-reverse** on page 231
- **ip rip receive-version** on page 233
- **ip rip send-version** on page 235
- **ip rip send default-also** on page 237
- **ip rip send default-only** on page 239
- **ip rip split-horizon** on page 241
- **show ip rip** on page 450

ip rip accept host-route

Configures the interface to accept host routes that neighbor routers advertise. A host route is one in which the destination IP address identifies a specific host, rather than a network or subnet. Use the no form of the command to disable host route acceptance.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

ip rip accept host-route

no ip rip accept host-route

Example

```
cli:# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli:# ip rip accept host-route
cli:# show ip rip
IP Address                          201.1.1.1
Send Version                            RIPv1
Receive Version                RIPv1 Or RIPv2
Cost                                        1
Authentication ON                       False
Authentication Type         No Authentication
Authentication Key Id                       0
Send Default Only                       False
Send Default Also                        True
Default Cost                                5
Accept Default                           True
Accept Host Route                        True
Split Horizon                            True
Poisoned Reverse                         True
Status                                 Active
```

Related Commands

- **ip rip accept default-route** on page 214
- **ip rip authentication** on page 218
- **ip rip cost** on page 221
- **ip rip default cost** on page 223
- **ip rip disable** on page 225
- **ip rip enable** on page 227
- **ip rip neighbor** on page 229
- **ip rip poisoned-reverse** on page 231
- **ip rip receive-version** on page 233
- **ip rip send-version** on page 235
- **ip rip send default-also** on page 237
- **ip rip send default-only** on page 239
- **ip rip split-horizon** on page 241
- **show ip rip** on page 450

ip rip authentication

Specifies the type of authentication (MD5 or simple password) that RIP uses as a security measure so that the current interface exchanges routing information with authorized neighbors only. This command also specifies the security key that is used for routing information exchanges. Make sure that:

- The Cuda 12000 and all of its RIP neighbors on the current interface use the same authentication type, key ID (for MD5 only), and key.
- You set the send version to RIP version 2 on the interface; otherwise, you cannot configure authentication. See the **ip rip send-version** description for more information.

Use the no form of the command to disable authentication.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

ip rip authentication {**md5** | **password** | **key-id** <id> **key** <key> | **key** <key>}

no ip rip authentication

Arguments

md5 — Specifies that the RIP interface uses MD5 authentication. MD5 authentication employs a shared secret key to generate a message digest, which is a 128-bit checksum of the packet and key. When a router sends a packet, it includes:

- The message digest
- A key ID that identifies the key that generated the message digest

At first, the receiving router accepts any packet that contains a key ID that matches one of its own key IDs. Using this key ID, the receiving router looks up the key. The receiving router uses the key associated with the received key ID to replace the message digest in the packet and generates a new message digest. If the generated message digest matches the message digest in the received packet, then the router accepts the packet. Otherwise, the router rejects the packet.

If you specify md5 as the authentication type, you must subsequently specify the shared secret key with the key-id <id> key <key> argument.

password — Specifies a simple password authentication method that includes a password in all RIP messages. When an interface receives a message that uses simple password authentication, the Cuda 12000 checks the incoming RIP message to see if the password is included. If the password is correct, the Cuda 12000 processes the message normally. If the password is not part of the incoming message, the Cuda 12000 drops the message.

If you specify password as the authentication type, you must subsequently specify the password with the key <key> argument.

key-id <id> **key** <key> — Specifies the md5 key ID and shared secret key:

- key-id <id> – An integer ranging from 1 to 255. This value identifies the shared secret key.
- key <key> – Specifies the key, which is a text string from 1 to 16 characters.

key <key> — Specifies the simple password, which is a text string from 1 to 16 characters.

Example 1

In this example, the user configures MD5 authentication:

```
cli:# ip rip authentication md5
cli:# ip rip authentication key-id 45 key my-secret-key
cli:#
```

Example 2

In this example, the user configures simple password authentication:

```
cli:# ip rip authentication password
cli:# ip rip authentication key secretky
cli:#
```

Related Commands

- **ip rip accept default-route** on page 214
- **ip rip accept host-route** on page 216
- **ip rip cost** on page 221
- **ip rip default cost** on page 223
- **ip rip disable** on page 225
- **ip rip enable** on page 227
- **ip rip neighbor** on page 229
- **ip rip poisoned-reverse** on page 231
- **ip rip receive-version** on page 233
- **ip rip send-version** on page 235
- **ip rip send default-also** on page 237
- **ip rip send default-only** on page 239
- **ip rip split-horizon** on page 241
- **show ip rip** on page 450

ip rip cost

Configures the cost or metric of the current interface. This cost is included in routes that originate on this interface, except the default route if a default cost is specified. The cost is a value that provides a general (but inexact) measure of the distance to a destination. Keep in mind that, when multiple routes to a destination exist, routers select the route with the lowest cost.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

ip rip cost <number>

Arguments

<number>	A number from 0 to 15. The default is 1.

Example

```
cli:# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli:# ip rip cost 3
cli:# show ip rip
IP Address                          201.1.1.1
Send Version                            RIPv1
Receive Version                RIPv1 Or RIPv2
Cost                                        3
Authentication ON                       False
Authentication Type         No Authentication
Authentication Key Id                       0
Send Default Only                       False
Send Default Also                       False
Default Cost                                0
Accept Default                           True
Accept Host Route                        True
Split Horizon                            True
Poisoned Reverse                         True
Status                                 Active
```

Related Commands

- **ip rip accept default-route** on page 214
- **ip rip accept host-route** on page 216
- **ip rip authentication** on page 218
- **ip rip default cost** on page 223
- **ip rip disable** on page 225
- **ip rip enable** on page 227
- **ip rip neighbor** on page 229
- **ip rip poisoned-reverse** on page 231
- **ip rip receive-version** on page 233
- **ip rip send-version** on page 235
- **ip rip send default-also** on page 237
- **ip rip send default-only** on page 239
- **ip rip split-horizon** on page 241
- **show ip rip** on page 450

ip rip default cost

Enter the cost or metric that is to be used for the default route entry in RIP updates originated on this interface. A value of zero indicates that no default route should be originated; in this case, a default route via another router may be propagated.

Before you can specify a default cost, you must first issue the **ip rip send default only** or **ip rip send default also** command.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

```
ip rip default cost <number>
```

Arguments

<number>	A number from 0 to 15. The default is 1.

Example

```
cli:# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli:# ip rip enable
cli:# ip rip send default-also
cli:# ip rip default cost 3
cli:# show ip rip
IP Address                          201.1.1.1
Send Version                            RIPv1
Receive Version                RIPv1 Or RIPv2
Cost                                        1
Authentication ON                       False
Authentication Type         No Authentication
Authentication Key Id                       0
Send Default Only                       False
Send Default Also                        True
Default Cost                                3
Accept Default                           True
Accept Host Route                        True
Split Horizon                            True
Poisoned Reverse                         True
Status                                 Active
```

Related Commands

- **ip rip accept default-route** on page 214
- **ip rip accept host-route** on page 216
- **ip rip authentication** on page 218
- **ip rip cost** on page 221
- **ip rip disable** on page 225
- **ip rip enable** on page 227
- **ip rip neighbor** on page 229
- **ip rip poisoned-reverse** on page 231
- **ip rip receive-version** on page 233
- **ip rip send-version** on page 235
- **ip rip send default-also** on page 237
- **ip rip send default-only** on page 239
- **ip rip split-horizon** on page 241
- **show ip rip** on page 450

ip rip disable

This command can be used to either disable an existing RIP interface or create a new RIP interface in the disabled state and use defaults for send-version and receive-version.

If the administrator uses this command to initially create a RIP interface then the send-version is defaulted to ripv1 compatible mode. That is, RIP transmits both RIPv1 and RIPv2 packets. Similarly receive-version is defaulted to both RIPv1 and RIPv2 mode. That is, RIP transmits both RIPv1 and RIPv2 packets.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

ip rip disable

Example

```
cli:# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli:# ip rip disable
cli:# show ip rip
IP Address                          201.1.1.1
Send Version                            RIPv1
Receive Version                RIPv1 Or RIPv2
Cost                                        1
Authentication ON                       False
Authentication Type         No Authentication
Authentication Key Id                       0
Send Default Only                       False
Send Default Also                        True
Default Cost                                3
Accept Default                           True
Accept Host Route                        True
Split Horizon                            True
Poisoned Reverse                         True
Status                        Not In Service
```

Related Commands

- **ip rip accept default-route** on page 214
- **ip rip accept host-route** on page 216
- **ip rip authentication** on page 218
- **ip rip cost** on page 221
- **ip rip default cost** on page 223
- **ip rip enable** on page 227
- **ip rip neighbor** on page 229
- **ip rip poisoned-reverse** on page 231
- **ip rip receive-version** on page 233
- **ip rip send-version** on page 235
- **ip rip send default-also** on page 237
- **ip rip send default-only** on page 239
- **ip rip split-horizon** on page 241
- **show ip rip** on page 450

ip rip enable

Enables Routing Information Protocol (RIP) on the current interface.

If send-version is not specified then the interface is configured with send-version defaulting to ripv1 compatible mode. That is, RIP transmits both RIPv1 and RIPv2 packets. If receive-version is not specified then the interface is configured with receive-version defaulting to both RIPv1 and RIPv2 mode. That is, RIP receives both RIPv1 and RIPv2 packets.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

ip rip enable

Example

```
cli:# ip rip enable
cli:# show ip rip
IP Address                          201.1.1.1
Send Version                            RIPv1
Receive Version                 RIPv1 Or RIPv2
Cost                                        1
Authentication ON                       False
Authentication Type          No Authentication
Authentication Key Id                       0
Send Default Only                       False
Send Default Also                        True
Default Cost                                3
Accept Default                           True
Accept Host Route                        True
Split Horizon                            True
Poisoned Reverse                         True
Status                                 Active
```

Related Commands

- **ip rip accept default-route** on page 214
- **ip rip accept host-route** on page 216
- **ip rip authentication** on page 218
- **ip rip cost** on page 221
- **ip rip default cost** on page 223
- **ip rip disable** on page 225
- **ip rip neighbor** on page 229
- **ip rip poisoned-reverse** on page 231
- **ip rip receive-version** on page 233
- **ip rip send-version** on page 235
- **ip rip send default-also** on page 237
- **ip rip send default-only** on page 239
- **ip rip split-horizon** on page 241
- **show ip rip** on page 450

ip rip neighbor

Configures the IP address of a RIP neighbor. Use the no form of the command to remove the RIP neighbor.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

ip rip neighbor <ip-address>

no ip rip neighbor <ip-address>

Arguments

<ip-address>	Specifies the IP address of the neighbor.

Example

```
cli:# interface 1/8/1
mode: interface:pos:csi(1/8/1)
cli:# ip address 20.1.1.1 255.255.255.0
cli:# ip rip enable
cli:# ip rip neighbor 20.1.1.2
cli:# show rip neighbors

row count: 1

RIP Interface   Neighbor IP     Neighbor   Last Update Status
IP Address      Address         Type       Received
--------------- --------------- ---------- ----------- --------
20.1.1.1        20.1.1.2        configured           0 Active
```

Related Commands

- **ip rip accept default-route** on page 214
- **ip rip accept host-route** on page 216
- **ip rip authentication** on page 218
- **ip rip cost** on page 221
- **ip rip default cost** on page 223
- **ip rip disable** on page 225
- **ip rip enable** on page 227
- **ip rip poisoned-reverse** on page 231
- **ip rip receive-version** on page 233
- **ip rip send-version** on page 235
- **ip rip send default-also** on page 237
- **ip rip send default-only** on page 239
- **ip rip split-horizon** on page 241
- **show ip rip** on page 450

ip rip poisoned-reverse

Configures the interface to implement poison reverse. Poison Reverse is a stronger form of split horizon. Routers do not omit destinations learned from an interface; instead, routers include these destinations in updates, but advertise an infinite cost to reach them. This parameter increases the size of routing updates. In addition, it provides a positive indication that a specific location is not reachable through a router.

The no form of the command disables poisoned reverse.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

ip rip poisoned-reverse

no ip rip poisoned-reverse

Example

```
cli:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli:# ip rip poisoned-reverse
cli:# show ip rip
IP Address                          201.1.1.1
Send Version                            RIPv1
Receive Version                RIPv1 Or RIPv2
Cost                                        1
Authentication ON                       False
Authentication Type         No Authentication
Authentication Key Id                       0
Send Default Only                       False
Send Default Also                        True
Default Cost                                3
Accept Default                           True
Accept Host Route                        True
Split Horizon                            True
Poisoned Reverse                         True
Status                                 Active
```

Related Commands

- **ip rip accept default-route** on page 214
- **ip rip accept host-route** on page 216
- **ip rip authentication** on page 218
- **ip rip cost** on page 221
- **ip rip default cost** on page 223
- **ip rip disable** on page 225
- **ip rip enable** on page 227
- **ip rip neighbor** on page 229
- **ip rip receive-version** on page 233
- **ip rip send-version** on page 235
- **ip rip send default-also** on page 237
- **ip rip send default-only** on page 239
- **ip rip split-horizon** on page 241
- **show ip rip** on page 450

ip rip receive-version

Specifies the version of RIP that you want this interface to use to learn RIP routes.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

ip rip receive-version {**none | 1 | 2 | 1 2**}

Arguments

none	Disables the ability to listen for incoming routes.
1	Enables RIP version 1.
2	Enables RIP version 2.
1 2	Enables both RIP version 1 and RIP version 2. This is the default.

Example

```
cli:# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli:# ip rip receive-version 1 2
cli:# show ip rip
IP Address                          201.1.1.1
Send Version                            RIPv1
Receive Version                RIPv1 Or RIPv2
Cost                                        1
Authentication ON                       False
Authentication Type         No Authentication
Authentication Key Id                       0
Send Default Only                       False
Send Default Also                        True
Default Cost                                3
Accept Default                           True
Accept Host Route                        True
Split Horizon                            True
Poisoned Reverse                         True
Status                                 Active
```

Related Commands

- **ip rip accept default-route** on page 214
- **ip rip accept host-route** on page 216
- **ip rip authentication** on page 218
- **ip rip cost** on page 221
- **ip rip default cost** on page 223
- **ip rip disable** on page 225
- **ip rip enable** on page 227
- **ip rip neighbor** on page 229
- **ip rip poisoned-reverse** on page 231
- **ip rip send-version** on page 235
- **ip rip send default-also** on page 237
- **ip rip send default-only** on page 239
- **ip rip split-horizon** on page 241
- **show ip rip** on page 450

ip rip send-version

Specifies the version of RIP that you want this interface to use to advertise routes.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

ip rip send-version {**none** | **1** | **2** | **1 2**}

Arguments

none	Disables RIP advertisements.
1	Enables RIP version 1.
2	Enables RIP version 2. This is the default.
1 2	Enables both RIP version 1 and RIP version 2.

Example

```
cli:# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli:# ip rip send-version 1 2
cli:# show ip rip
IP Address                          201.1.1.1
Send Version                 RIPv1 Compatible
Receive Version                RIPv1 Or RIPv2
Cost                                        1
Authentication ON                       False
Authentication Type         No Authentication
Authentication Key Id                       0
Send Default Only                        True
Send Default Also                       False
Default Cost                                3
Accept Default                           True
Accept Host Route                        True
Split Horizon                            True
Poisoned Reverse                         True
Status                                 Active
```

Related Commands

- **ip rip accept default-route** on page 214
- **ip rip accept host-route** on page 216
- **ip rip authentication** on page 218
- **ip rip cost** on page 221
- **ip rip default cost** on page 223
- **ip rip disable** on page 225
- **ip rip enable** on page 227
- **ip rip neighbor** on page 229
- **ip rip poisoned-reverse** on page 231
- **ip rip receive-version** on page 233
- **ip rip send default-also** on page 237
- **ip rip send default-only** on page 239
- **ip rip split-horizon** on page 241
- **show ip rip** on page 450

ip rip send default-also

Configures the interface to advertise the default route in addition to other routes. Use the no form of the command to disable default route advertisement on the interface.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

ip rip send default-also

no ip rip send default-also

Example

```
cli:# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli:# ip rip send default-also
cli:# show ip rip
IP Address                          201.1.1.1
Send Version                            RIPv1
Receive Version                RIPv1 Or RIPv2
Cost                                       1
Authentication ON                      False
Authentication Type         No Authentication
Authentication Key Id                      0
Send Default Only                      False
Send Default Also                       True
Default Cost                               3
Accept Default                          True
Accept Host Route                       True
Split Horizon                           True
Poisoned Reverse                        True
Status                                Active
```

Related Commands

- **ip rip accept default-route** on page 214
- **ip rip accept host-route** on page 216
- **ip rip authentication** on page 218
- **ip rip cost** on page 221
- **ip rip default cost** on page 223
- **ip rip disable** on page 225
- **ip rip enable** on page 227
- **ip rip neighbor** on page 229
- **ip rip poisoned-reverse** on page 231
- **ip rip receive-version** on page 233
- **ip rip send-version** on page 235
- **ip rip send default-only** on page 239
- **ip rip split-horizon** on page 241
- **show ip rip** on page 450

ip rip send default-only

Configures the interface to advertise the default route only. The interface advertises no other routes. Use the no form of the command to disable advertising the default route only on the interface.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

ip rip send default-only

no ip rip send default-only

Example

```
cli:# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli:# ip rip send default-only
cli:# show ip rip
IP Address                          201.1.1.1
Send Version                            RIPv1
Receive Version                RIPv1 Or RIPv2
Cost                                        1
Authentication ON                       False
Authentication Type         No Authentication
Authentication Key Id                       0
Send Default Only                        True
Send Default Also                       False
Default Cost                                3
Accept Default                           True
Accept Host Route                        True
Split Horizon                            True
Poisoned Reverse                         True
Status                                 Active
```

Related Commands

- **ip rip accept default-route** on page 214
- **ip rip accept host-route** on page 216
- **ip rip authentication** on page 218
- **ip rip cost** on page 221
- **ip rip default cost** on page 223
- **ip rip disable** on page 225
- **ip rip enable** on page 227
- **ip rip neighbor** on page 229
- **ip rip poisoned-reverse** on page 231
- **ip rip receive-version** on page 233
- **ip rip send-version** on page 235
- **ip rip send default-also** on page 237
- **ip rip split-horizon** on page 241
- **show ip rip** on page 450

ip rip split-horizon

Configure the interface to implement split horizon. Split horizon specifies that if a router learns a route from an update received on the interface, then the router does not advertise that route on updates that it transmits on the interface.

Use the no form of the command to disable split horizon on the interface.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

ip rip split-horizon

no ip rip split-horizon

Example

```
cli:# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli:# ip rip split-horizon
cli:# show ip rip
IP Address                          201.1.1.1
Send Version                 RIPv1 Compatible
Receive Version                RIPv1 Or RIPv2
Cost                                        1
Authentication ON                       False
Authentication Type          No Authentication
Authentication Key Id                       0
Send Default Only                        True
Send Default Also                       False
Default Cost                                3
Accept Default                           True
Accept Host Route                        True
Split Horizon                            True
Poisoned Reverse                         True
Status                                 Active
```

Related Commands

- **ip rip accept default-route** on page 214
- **ip rip accept host-route** on page 216
- **ip rip authentication** on page 218
- **ip rip cost** on page 221
- **ip rip default cost** on page 223
- **ip rip disable** on page 225
- **ip rip enable** on page 227
- **ip rip neighbor** on page 229
- **ip rip poisoned-reverse** on page 231
- **ip rip receive-version** on page 233
- **ip rip send-version** on page 235
- **ip rip send default-also** on page 237
- **ip rip send default-only** on page 239
- **show ip rip** on page 450

ip route

Adds a static route to the routing table on the current interface. A static route is a manually configured route that specifies the network path that a packet must follow. You can configure static routes to make packets follow specifically configured paths. Static routes remain in the IP routing table until you remove them using the **no ip route** command. To display the routing table, use the **show ip** command. Static routes are identified in the routing table as protocol type "Net Mgmt."

Mode

```
interface:<type>:csi:<c/s/i>
```

Syntax

```
ip route <dest-network> <mask> <gateway-ip-address> [<metric>]

no ip route <dest-network> <mask>
```

Arguments

<dest-network>	Destination network IP address.
<mask>	Destination network mask.
<gateway-ip-address>	IP address of the gateway for this route.
<metric>	Metric for this route. The metric is a number that is used to select the route when multiple routes to the same destination exist. The route with the lowest metric is selected. The default is 1.

Example 1

Adding a static route:

```
cli:192.168.208.3:root# ip route 201.1.8.0 255.255.255.0 201.1.1.5
cli:192.168.208.3:root# show ip

row count: 4

Protocol Route  Destination     Net Mask        Next Hop        Metric C/S/I
         Type
-------- ------ --------------- --------------- --------------- ------ ------
Local    Local  172.16.19.0     255.255.255.0   172.16.19.89         0 1/11/4
Local    Local  201.1.1.0       255.255.255.0   201.1.1.1            0 1/1/1
Local    Local  201.1.2.0       255.255.255.0   201.1.2.1            0 1/1/1
Net Mgmt Remote 201.1.8.0       255.255.255.0   201.1.1.5            1 1/1/1
```

Example 2

Removing a static route:

```
cli:192.168.208.3:root# show ip
row count: 4
Protocol Route  Destination     Net Mask        Next Hop        Metric C/S/I
         Type
-------- ------ --------------- --------------- --------------- ------ ------
Local    Local  172.16.19.0     255.255.255.0   172.16.19.89         0 1/11/4
Local    Local  201.1.1.0       255.255.255.0   201.1.1.1            0 1/1/1
Local    Local  201.1.2.0       255.255.255.0   201.1.2.1            0 1/1/1
Net Mgmt Remote 201.1.8.0       255.255.255.0   201.1.1.5            1 1/1/1

cli:192.168.208.3:root# no ip route 201.1.8.0 255.255.255.0
cli:192.168.208.3:root# show ip
row count: 3
Protocol Route  Destination     Net Mask        Next Hop        Metric C/S/I
         Type
-------- ------ --------------- --------------- --------------- ------ ------
Local    Local  172.16.19.0     255.255.255.0   172.16.19.89         0 1/11/4
Local    Local  201.1.1.0       255.255.255.0   201.1.1.1            0 1/1/1
Local    Local  201.1.2.0       255.255.255.0   201.1.2.1            0 1/1/1
```

Related Commands

- **ip route default** on page 245
- **show ip** on page 438

ip route default

Configures the default IP route, which is selected to forward traffic when a specific route to a destination does not exist. Use the no form of the command to remove the default IP route.

Mode

Any.

Syntax

ip route default <gateway-ip-address> [<metric>]

no ip route default

Arguments

<gateway-ip-address>	Specifies the IP address of the gateway to which the Cuda 12000 will forward packets when no specific route to a destination exists.
<metric>	Metric for the default route. The metric is a number that is used to select the route when multiple routes to the same destination exist. The route with the lowest metric is selected. The default is 1.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# ip route default 201.1.1.2
cli:192.168.208.3:interface:cable:csi(1/1/1)# show ip
row count: 4
Protocol Route  Destination     Net Mask        Next Hop        Metric C/S/I
         Type
-------- ------ -------------- -------------- -------------- ------ ------
Net Mgmt Remote 0.0.0.0         0.0.0.0         201.1.1.2            1 1/1/1
Local    Local  201.1.1.0       255.255.255.0   201.1.1.1            0 1/1/1
Local    Local  201.1.2.0       255.255.255.0   201.1.2.1            0 1/1/1
```

Related Commands

- **ip route** on page 243
- **show ip** on page 438

ip source-route

This command allows you to configure a default route (next hop gateway) to which a packet containing a particular source IP address should be forwarded in the event that a local route to the destination does not exist.

Use the no form of the command to remove a source route entry from the current interface.

Mode

```
interface:<type>:csi(<c/s/i>)
```

Syntax

```
ip source-route <ip-address> <mask> <next-hop-gateway>

no ip source-route <ip-address> <mask> <next-hop-gateway>
```

Arguments

<ip-address>	IP address of the source route that you want to match against.
<mask>	IP mask of the source route that you want to match against.
<next-hop-gateway>	Next hop gateway to forward all packets with matching source routes in the event a default route does not exist in the routing table.

Example

The following example configures interface 1/1/1 to forward any packets received from the 172.16.19.0 network to the router at 172.20.19.4:

```
cli:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:interface:cable:csi(1/1/1)# ip source-route 172.16.19.0 255.255.255.0
172.20.19.4
```

Related Commands

- **show ip interface source-route** on page 447

loop

Configures the current interface to perform loopback testing. Loopback testing allows you to verify the connection between the current POS interface and a remote device.

The loopback test allows a ping packet to loop through the remote device and current interface. If the ping packet completes the loop, the connection is sound; inability to complete the loop indicates an equipment malfunction within the connection path.

The interface supports two kinds of loopback verification:

- **Line** — Configures the interface to loopback data back to the remote router that sent it. Incoming data is received and then looped back to the sending router without actually transmitting the packets. This form of loopback is helpful in verifying the connection between the current interface and a remote router.
- **Internal** — Also known as local loopback, configures the interface to loopback outgoing data that it originates without actually transmitting it over the network. This version of loopback is helpful in verifying that the POS interface is operational.

By default, no form of loopback is configured on the interface. Use the **no loop** command to remove loopback configuration on the current interface.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
loop {line | internal}
```

Arguments

{**line** | **internal**} Specifies one of the following values:

- line – Configures the current interface for line loopback verification.
- internal – Configures the current interface for internal (local) loopback verification.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# loop line
cli#
```

Related Commands

- **interface** on page 181
- **no loop** on page 275

link-trap

Enables link up and link down traps for an interface. Use the no form of the command to disable link up and link down traps for an interface.

Mode

```
interface:<type>:<c/s/i>
```

Syntax

link-trap

no link-trap

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# link-trap
cli:192.168.208.3:interface:cable:csi(1/1/1)# show link-trap
Link Up/Down Trap Enable             Enabled

cli:192.168.208.3:interface:cable:csi(1/1/1)# no link-trap
cli:192.168.208.3:interface:cable:csi(1/1/1)# show link-trap
Link Up/Down Trap Enable            Disabled
```

Related Commands

- **show link-trap** on page 452

lookup

Enables and disables the Jini lookup service on the Cuda 12000. Jini provides a simple platform-independent environment for delivering network services and for enabling interaction between programs that use these services. Note that at least one Jini lookup service should be running on the attached network in order to support multi-chassis groups. Two lookup services are recommended (redundancy).

Mode

`root`

Syntax

`lookup {enable | disable}`

Arguments

{**enable | disable**} Specifies one of the following values:

- enable – Enables the Jini lookup service.
- disable (default) – Disables the Jini lookup service.

Example

```
cli:192.168.208.3:root# lookup disable
# rmid (pid 840 781 776 773 770 769 767 766 764 762 761 760 718 717 716 701
699
696 694 545) is running...
# Stopping RMI activation daemon:
cli:192.168.208.3:root# lookup enable
Please wait, this may take some time ...
# rmid is stopped
# Starting RMI activation daemon: OK
#
# JINI lookup service (reggie) is running.
cli:192.168.208.3:root# show lookup
```

Related Commands

- **chassis** on page 77
- **show lookup** on page 453

map-list

Adds a route-map to a specified map-list. The map-list is a sequential listing of route-maps used to filter incoming or outgoing OSPF and RIP routes.

Use the **no map-list** command to remove the configuration.

Mode

To create an import OSPF map list:

```
router:ospf:import
```

To create an export OSPF map list:

```
router:ospf:export
```

To create an import RIP map list:

```
router:rip:import
```

To create an export RIP map list:

```
router:rip:export
```

Syntax

```
map-list <route-map-list-number> route-map <route-map-number>
set {active | inactive}
```

Arguments

<route-map-list-number>	Number of the map-list to which you want to add a route-map.
route-map <route-map-number>	Number of the route map that you want to add.
active	Sets the map-list to the active state. The system will then use the map-list to filter routes. There can only be a single active map-list at one time.
inactive	Sets the map-list to an inactive state.

Example

```
cli:172.16.19.10:root# router rip import
mode: router:rip:import
cli:172.16.19.10:router:rip:import# map-list 1 route-map 1
cli:172.16.19.10:router:rip:import# map-list 1 route-map 2
cli:172.16.19.10:router:rip:import# map-list 1 set active
cli:172.16.19.10:router:rip:import# show map-list 1

row count: 2

Template Template Row Status
         Order
-------- -------- --------------
       1        1 Active
       2        2 Active

cli:172.16.19.10:router:rip:import#
```

Related Commands

- **export** on page 166
- **import** on page 179
- **override** on page 309
- **route-map** on page 354
- **show map-list** on page 456
- **show route-map** on page 517

map-timer

Sets the map timer interval, in microseconds, on a specific cable interface.

Mode

```
interface <type> c/s/i
```

Syntax

```
map-timer <value>
```

Arguments

<value>	CMTS hardware Map timer period interval, in microseconds. Valid range: 1000–10000 microseconds. Default: 10 milliseconds.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# map-timer 1000
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **map-list** on page 251
- **route-map** on page 354
- **show interface cable <c/s/i> mac** on page 421
- **show mac** on page 454

match

Creates match attributes for import and export route filters.

Mode

```
router:ospf:import:route-map()
router:ospf:export:route-map()
router:rip:import:route-map()
router:rip:export:route-map()
```

Syntax

```
match ip-address <ip address> <mask> tag <tag-value>
{exact | exclude | peer-address <ip address> peer-mask <mask>}
```

Arguments

ip-address <ip address>	Specify the route address used to match against route entries.
<mask>	Specify the route mask used to match against route entries.
tag <tag-value>	Specify the route tag used to match against route entries.
exact	If the tag field is specified and exact is specified then the match is intended to match all routes with an exact match on the tag field.
exclude	If the tag field is specified and exclude is specified then the match is intended to match all routes that do not match on the tag field.
peer-address <ip address>	Specify the source or destination router ip address, depending upon if this match is used for an import of export filter.
peer-mask <mask>	Specify the source or destination router ip address mask, depending upon if this match is used for an import of export filter.

Example

```
cli:172.16.19.11:root# router ospf import
mode: router:ospf:import
cli:172.16.19.10:router:ospf:import# route-map 10
cli:172.16.19.10:router:ospf:import:route-map(10)# match tag 124 exact
cli:172.16.19.10:router:ospf:import:route-map(10)# match ip-address 172.16.19.1
0 255.255.255.255
cli:172.16.19.10:router:ospf:import:route-map(10)# show route-map 10
ID                                        10
Description
Route Address                    172.16.19.10
Route Mask                   255.255.255.255
Peer Address                       0.0.0.0
Peer Mask                          0.0.0.0
Tag                                    124
Key Bits                               529
Preference                               0
Flags                                    1

cli:172.16.19.10:router:ospf:import:route-map(10)#
```

Related Commands

- **map-list** on page 251
- **route-map** on page 354
- **show route-map** on page 517

modulation-profile

Modulation profiles contain the burst profile properties for the Cuda 12000 upstream data stream channels. By default, the Cuda 12000 supports two profiles for the four upstream channels (for 1x4 modules) or six upstream channels (for 1x6 modules). Each modulation profile defines a burst descriptor for the following Interval Usage Codes:

- **Request** — Interval when a request on bandwidth can be sent by the modem.
- **Initial Maintenance** — Interval when new modems can start establishing a connection to the CMTS with Initial Ranging Requests.
- **Station Maintenance** — Interval when modems perform periodic ranging with periodic ranging fro adjusting power, timing and frequency.
- **Short Data** — Interval when a modem can send an upstream PDU which is shorter than the defined maximum burst size.
- **Long Data** — Interval when a modem can send a larger data PDU.

Mode

```
root
interface <type> c/s/i
```

Syntax

```
modulation-profile <profile number>
interval-usage {initial | long | request | short | station}
[fec-tbytes <number>]
[fec-len <number>]
[burst-len <number>]
[mod {16qam | qpsk}]
[scrambler]
[no scrambler]
[diff]
[no diff]
[seed <number>]
[pre-len <number>]
[last-cw {fixed | shortened}]
```

Arguments

<profile-number>	The profile number identifies this modulation profile.
Interval-usage	Valid usage codes are initial, short, long, request, and station.
fec-tbytes <number>	Specify the number of FEC Error Correction bytes. The number of bytes ranges from 0 to 10, where 0 implies no FEC.
fec-len <number>	The number of data bytes (k) in the forward error correction codeword. Acceptable values range from 8 to 255.
burst-len <number>	The maximum number of mini-slots that can be transmitted during a channel's burst time. A value of zero is transmitted if the burst length is bounded by the allocation MAP rather than this profile. The valid range is 0 to 256.
mod {**16qam** | **qpsk**}	This is the modulation rate for an upstream port. Mod can be set to QPSK or QAM16.
scrambler	This enables the scrambler. Scrambler is disabled by default.
no scrambler	This disables the scrambler.
diff	Enable differential encoding.
no diff	Disable differential encoding.
seed <number>	Specify the 15 bit seed value for the scrambler polynomial. Valid values are 0 to 32767. The default is 338.
pre-len <number>	Specify the preamble pattern length from 2 to 448 bits.
last-cw	Specify the code word type as fixed or shortened.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# modulation-profile 7 interval-usage
initial fec-len 8
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **no modulation-profile** on page 276
- **show interface cable <c/s/i> modulation-profile** on page 425

mtu

Sets the maximum transmission unit (MTU), in bytes, on the current POS interface. The default MTU transmission size is 1500 bytes. Use the **no mtu** command to restore this default.

Note that only the default 1500-byte MTU size is currently supported. Do not change this value, as there is no support for fragmented packets.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
mtu <value>
```

Arguments

<value>	MTU size, in bytes. Valid range: 64 – 1500. Default: 1500.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# mtu 1200
cli#
```

Related Commands

- **interface** on page 181
- **no mtu** on page 277
- **show interface pos** on page 437

negotiation auto

Configures an Ethernet port to automatically negotiate duplex mode and speed. If you had explicitly configured the duplex mode for full or half, or explicitly configured speed for 10 Mbps or 100 Mbps, then these settings are no longer in effect.

Use the no form of the command to disable the ability to automatically negotiate duplex mode and speed. If you had previously used the **duplex** and **speed** commands to configure the ability to automatically negotiate duplex and speed, the **no negotiation auto** command resets the duplex mode and speed to their previous explicit settings.

Mode

```
interface:ethernet:csi(<c/s/i>)
```

Syntax

```
negotiation auto

no negotiation auto
```

Example

```
cli:192.168.208.3:interface:ethernet:csi(1/11/1)# negotiation auto
cli:192.168.208.3:interface:ethernet:csi(1/11/1)# show interface
Interface                  1/11/1 Ethernet (100 Mb)
Configured duplex                        auto
Configured speed                         auto
Auto-negotiation                       enable
In octets                                   0
In unicast                                  0
In multicast                                0
In broadcast                                0
Out octets                                  0
Out unicast                                 0
Out multicast                               0
Out broadcast                               0
```

Related Commands

- **duplex** on page 156
- **show interface ethernet** on page 435
- **speed** on page 582

no access-class

Deletes the specified access-class from the current cable interface.

Mode

`interface:cable:csi(<c/s/i>)`

Syntax

no access-class <number> {**in** | **out**} **priority** <priority number>

Arguments

<number>	Number of the access-class that you want to remove.
in	Specifies an access class applied to the inbound interface.
out	Specifies an access class applied to the outbound interface.
priority <priority number>	Priority number of the access-class that you want to remove.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# show access-class

row count: 1

Access List      Priority Row
List   Direction          Status
Number
------ --------- -------- ------
     1        in        1      1

cli:172.16.19.10:interface:cable:csi(1/1/1)# no access-class 1 in priority 1
cli:172.16.19.10:interface:cable:csi(1/1/1)# show access-class


Access List      Priority Row
List   Direction          Status
Number
------ --------- -------- ------

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **access-class** on page 45
- **show access-class** on page 369

no access-list

Deletes the specified access-list.

Mode

root

Syntax

no access-list {<number> | **all**}

Arguments

<number>	Specify the number of a specific access-list that you want to delete.
all	Use this keyword to delete all access lists configured within the system.

Example

```
cli:172.16.19.10:root# show access-list

row count: 2

List  Ac Rule Prot IP Source       Mask Source     Start End   Estab TOS ChTOS
                   IP Dest         Mask Dest       Port  Port  Sync  Msk
----- -- ---- ---- --------------- --------------- ----- ----- ----- --- ------
    1 DE    1  tcp 0.0.0.0         255.255.255.255    23    23 False   0      0
                   0.0.0.0         255.255.255.255     0 65535 False 255
..... .. .... .... ............... ............... ..... ..... ..... ... ......
    1 PE    2   ip 0.0.0.0         255.255.255.255            False   0      0
                   0.0.0.0         255.255.255.255            False 255
..... .. .... .... ............... ............... ..... ..... ..... ... ......

cli:172.16.19.10:root# no access-list 1
cli:172.16.19.10:root# show access-list
cli:172.16.19.10:root#
```

Related Commands

- **access-list** on page 47
- **show access-list** on page 370

no asbr

This global Open Shortest Path First (OSPF) command disables Autonomous System Border Router functionality for the router.

Mode

```
router:ospf
```

Syntax

```
no asbr
```

Example

```
cli:172.16.19.10:root# router ospf
mode: router:ospf
cli:172.16.19.10:router:ospf# show ospf | include ASBR
ASBR Status                              True
cli:172.16.19.10:router:ospf# no asbr
cli:172.16.19.10:router:ospf# show ospf | include ASBR
ASBR Status                             False

cli:172.16.19.10:router:ospf#
```

Related Commands

- **asbr** on page 54
- **show ospf** on page 489

no bootp-policy

Removes the specified BOOTP policy.

Mode

root

Syntax

no bootp-policy {<number> | **all**}

Arguments

<number>	Number of the specific BOOTP policy that you want to remove from the current cable interface.
all	Use this keyword to delete all BOOTP policies from the current cable interface.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# show bootp-policy 1
Index                                    1
Description                policy_deny
Mac Address
Mask
Action                                 deny
Server List

cli:172.16.19.10:interface:cable:csi(1/1/1)# no bootp-policy 1
cli:172.16.19.10:interface:cable:csi(1/1/1)# show bootp-policy 1
Bootp policy 1 does not exist.
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **bootp-policy** on page 68
- **show bootp-policy** on page 379

no clock-source

Configures the POS interface to obtain its clock source from the incoming receive network (the default behavior*)*. The system uses this clock to synchronizing SONET transmission.

Use the **clock-source** command to configure the SONET transmission clock-source for the current interface.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
no clock-source
```

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# no clock-source
cli#
```

Related Commands

- **clock-source** on page 87
- **interface** on page 181
- **show interface pos** on page 437

no crc

Configures the current POS interface to use the default 32-bit CRC error checking. Use the **crc** command to configure error checking on the current interface. For more information about CRC error checking, see the **crc** command on page 134.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
no crc
```

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# no crc
cli#
```

Related Commands

- **crc** on page 134
- **interface** on page 181
- **show interface pos** on page 437

no db-connect

Removes the provision.cfg file which contains information required to access the provisioning database. Note that the /bas/data/provision/provision.cfg file is immediately deleted after issuing this command.

Use this command with caution, as it removes the information that FastFlow Broadband Provisioning Manager applications require to access the provisioning (LDAP) database. After issuing this command, you must use the **db-connect** command to specify new access information to the LDAP database.

Mode

`root`

Syntax

`no db-connect`

Related Commands

- **db-connect** on page 136
- **show db-connect** on page 398

no dhcp-relay

Use this command to remove the following DHCP-relay configuration on a specified cable interface:

- Host (CPE) gateway
- Cable modem (CM) gateway
- MTA gateway
- DHCP server address

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
no dhcp-relay {cm-gateway | cpe-gateway | mta-gateway |
  server <ip-address>}
```

Arguments

cpe-gateway	Deletes the CPE host gateway on the current cable interface.
cm-gateway	Deletes the cable modem gateway on the current cable interface.
mta-gateway	Deletes the MTA gateway on the current cable interface.
server <ip-address>	Removes the DHCP server address from the current cable interface, after which the interface no longer forwards DHCP requests to the server.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# no dhcp-relay cm-gateway
cli:172.16.19.10:interface:cable:csi(1/1/1)# no dhcp-relay cpe-gateway
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **dhcp-relay** on page 145
- **show dhcp-relay** on page 402

no enable

Use this command to log out of the Cuda 12000 command line interface (CLI).

Mode

Any.

Syntax

`no enable`

Example

The following example uses the `no enable` command to log out of the CLI then uses the `enable` command to log back into it:

```
cli:172.16.19.10:root# no enable
Disabling login session...
cli:172.16.19.10:root> enable root
password: ***
Looking up: //172.16.19.10:1099/BasJavaServer...
Java Server version is compatible
ClientMode: CLI
logon complete
cli:172.16.19.10:root#
```

Related Commands

- **enable** on page 158

no ip address

Removes an IP address from the current interface.

Mode

```
interface:<type>:csi(<c/s/i>)
```

Syntax

```
no ip address <ip-address>
```

Arguments

<ip-address>	IP address that you want to remove from the current interface.

Example

```
cli:172.16.19.10:root# interface 1/11/1
mode: interface:ethernet:csi(1/11/1)
cli:172.16.19.10:interface:ethernet:csi(1/11/1)# show ip address
Chassis/Slot/Interface      1/11/1

row count: 1

IP Address       Net Mask         If Index   Priority
---------------- ---------------- ---------- ----------
209.240.193.4    255.255.255.0      11337729    Primary

cli:172.16.19.10:interface:ethernet:csi(1/11/1)# no ip address 209.240.193.4
cli:172.16.19.10:interface:ethernet:csi(1/11/1)# show ip address
Chassis/Slot/Interface      1/11/1

row count: 0

IP Address       Net Mask         If Index   Priority
---------------- ---------------- ---------- ----------

cli:172.16.19.10:interface:ethernet:csi(1/11/1)#
```

Related Commands

- **ip address** on page 183

no ip igmp

Removes IGMP settings for an interface. To list settings, issue the **show ip igmp** command.

Mode

`interface`

Syntax

```
no ip igmp {join-group <group-address> | query-interval |
  query-max-response-time | version | robustness | router |
  last-query-interval}
```

Arguments

join-group <group-address>	Terminates the Cuda 12000's membership in the specified multicast group.
query-interval	Removes the current query interval and resets this parameter to the default (125 seconds).
query-max-response-time	Removes the current query maximum response time and resets this parameter to the default (10 seconds).
version	Removes the current IGMP version and resets this parameter to the default (Version 2).
robustness	Removes the current robustness value and resets this parameter to the default (2).
router	Disables the IGMP Querier (router) capability on the interface. **Note**: The DOCSIS module can only be configured as an IGMP querier (router).
last-query-interval	Removes the current last query interval value and resets this parameter to the default (1 second). Note that this parameter does not apply to IGMP Version 1.

Example

In this example, the user terminates membership in the multicast group 225.2.2.1:

```
cli:interface:ethernet:csi(1/3/1)# no ip igmp join-group 225.2.2.1
```

Related Commands

- **ip igmp** on page 187
- **show ip** on page 438

no ip igmp proxy

Disables the ability to proxy for the specified multicast address or address range. If you need to know the list of proxies, issue the **show ip igmp proxy** command.

Mode

`root`

Syntax

`no ip igmp proxy` `<group-address> <group-mask>` **`metric`** `<number>`

Arguments

<group-address>	Class D IP address of the multicast group (for example, 239.1.1.1) for which the Cuda 12000 no longer proxies.
<group-mask>	A mask, in dot-notation format, that specifies a multicast address range.
metric <number>	Metric value from 1 to 255 that assigns a priority to the proxy.

Example

In this example, the user disables the ability to proxy for multicast group 225.4.3.4:

```
cli:192.168.220.208:root# no ip igmp proxy 225.4.3.4 255.255.255.255 metric 5
cli:192.168.220.208:root#
```

Related Commands

- **ip igmp proxy** on page 189
- **show ip {mroute | igmp}** on page 439

no ip rip

Removes RIP capabilities from an IP interface.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address)
```

Syntax

```
no ip rip
```

Example

```
cli:interface:cable:csi(1/1/1):ip-address(201.1.1.1) no ip rip
```

Related Commands

- **ip rip accept default-route** on page 214
- **ip rip accept host-route** on page 216
- **ip rip authentication** on page 218
- **ip rip cost** on page 221
- **ip rip default cost** on page 223
- **ip rip disable** on page 225
- **ip rip enable** on page 227
- **ip rip neighbor** on page 229
- **ip rip poisoned-reverse** on page 231
- **ip rip receive-version** on page 233
- **ip rip send-version** on page 235
- **ip rip send default-also** on page 237
- **ip rip send default-only** on page 239
- **ip rip split-horizon** on page 241
- **show ip rip** on page 450

no loop

Turns off loopback testing on the current interface. You can use the `loop` command to enable and configure loopback testing.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
no loop {internal | line}
```

Arguments

internal	Disables internal loopback testing.
line	Disables line loopback testing.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# no loop internal
cli#
```

Related Commands

- **interface** on page 181
- **loop** on page 247
- **show interface pos** on page 437

no modulation-profile

Deletes a modulation profile. A modulation profile can only be deleted if it is not in use by any upstream channel. Also note that modulation profiles 1 and 2 are CMTS default modulation profiles and cannot be deleted.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

no modulation-profile <profile-number>

Arguments

<profile-number> Index number identifying the modulation profile that you want to remove.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# show modulation-profile 3

row count: 3

Interval Mod   Pream  Pream Diff FEC   FEC CW Scram Max   Guard Last Scram
Usage    Type  Offset Len        Error Len    Seed  Burst Time  CW   Enable
                                                    Size
-------- ----- ------ ----- ---- ----- ------ ----- ----- ----- ---- ------
Request  QPSK     504   384  yes     5     34     0     0     8  no    yes
Short    QPSK     496   384  yes     5     34     0     0     8  no    yes
Long     QPSK     488   384  yes     5     34     0     0     8  no    yes

cli:172.16.19.10:interface:cable:csi(1/1/1)# no modulation-profile 3
terface:cable:csi(1/1/1)# show modulation-profile 3
Modulation Profile 3 does not exist!
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **modulation-profile** on page 256
- **show interface cable <c/s/i> modulation-profile** on page 425

no mtu

Sets the maximum transmission unit (MTU) size for the current interface back to the default MTU size of 1500 bytes.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
no mtu
```

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# no mtu
cli#
```

Related Commands

- **interface** on page 181
- **mtu** on page 258
- **show interface pos** on page 437

no ospf area

Removes an OSPF area from the system. If you specify the authentication argument, you disable authentication for the area.

Mode

```
router:ospf
```

Syntax

```
no ospf area <area id> [authentication]
[range <ip-address> <mask> [advertise-matching]]
```

Arguments

<area-id>	Area ID of the area that you want to delete from the system.
authentication	Disables authentication for the OSPF area. For example, if the area was configured for MD5 authentication, this argument would disable authentication for the area.
range <ip-address> <mask> [**advertise-matching**]	Removes a summary range.

Example

```
cli:172.16.19.10:root# router ospf
mode: router:ospf
cli:172.16.19.10:router:ospf# show ospf area

row count: 1

Area Id          Auth Type ImpAsExt SPF  ABR   ASBR  LSA   LSA    Area    Area
                                        Count Count Count Cksum  Summary Type
---------------- --------- -------- ---- ----- ----- ----- ------ ------- -----
2.2.2.2               None   Extern    1     0     0     1  56944    Send  Norm

cli:172.16.19.10:router:ospf# no ospf area 2.2.2.2
cli:172.16.19.10:router:ospf# show ospf area


Area Id          Auth Type ImpAsExt SPF  ABR   ASBR  LSA   LSA    Area    Area
                                        Count Count Count Cksum  Summary Type
---------------- --------- -------- ---- ----- ----- ----- ------ ------- -----

cli:172.16.19.10:router:ospf#
```

Related Commands

- **ospf area** on page 307
- **router ospf** on page 359

no pos flag

Use this command to remove the setting of the following overhead byte values:

- **c2** — This byte serves as a signal label and indicates the content held within the synchronous payload envelope (SPE) of the SONET frame, as well as the status of that payload.
- **J0** — The section trace byte required for interoperability with some SDH devices in Japan.

Use the **pos flag** command to configure these SONET overhead byte values.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
no pos flag {c2 | j0} <value>
```

Arguments

c2 <value>	Path signal identifier used to identify the payload content type. Valid range: 0 – 255.
j0 <value>	Section trace byte. For interoperability with SDH equipment. Valid range: 0 – 255.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# no pos flag c2
cli#
```

Related Commands

- **interface** on page 181
- **pos flag** on page 316
- **show interface pos** on page 437

no pos report

Disables logging of select SONET alarms. For more information about SONET alarms and reporting, see the **pos report** command.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
no pos report {lais|lrdi|pais|plop|prdi|sd-ber|sf-ber|slof|slos}
```

Arguments

lais	Disables line alarm indication signal error reporting.
lrdi	Disables report line remote defect indication error reporting.
pais	Disables path alarm indication signal error reporting.
plop	Disables path loss of pointer error reporting.
prdi	Disables path remote defect indication error reporting.
sd-ber	Disables signal degradation BER error reporting.
sf-ber	Disables signal failure BER error reporting.
slof	Disables section loss of frame error reporting.
slos	Disables section loss of signal error reporting.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# no pos report lais
cli#
```

Related Commands

- **interface** on page 181
- **pos report** on page 317
- **show interface pos** on page 437

no pos scramble

Disables payload scrambling on the current packet over SONET (POS) interface. To enable scrambling, use the **pos scramble** command. For more information about payload scrambling, see the **pos scramble** command.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
no pos scramble
```

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# no pos scramble
cli#
```

Related Commands

- **interface** on page 181
- **pos scramble** on page 320
- **show interface pos** on page 437

no pos threshold

Use this command to set the signal degrade Bit Error Rate (sd-BER) and signal fail Bit Error Rate (sf-BER) thresholds back to their default values.

The default threshold value for the signal degrade BER is 6; default for Signal failure BER is 3. When these thresholds are reached, the system sends the corresponding SONET alarm to the POS controller.

You can view the defects that have been reported using the `show controllers` command. To configure these threshold values, use the **pos threshold** command.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
no pos threshold {sd-ber | sf-ber}
```

Arguments

sd-ber	Sets the threshold for signal degradation BER errors on the current interface back to the default value of 6.
sf-ber	Sets the threshold for report signal failure BER errors on this interface back to the default value of 3.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# no pos threshold sf-ber
cli#
```

Related Commands

- **interface** on page 181
- **pos threshold** on page 321
- **show interface pos** on page 437

no ppp authentication

Disables Point-to-Point protocol (PPP) authentication on the current SONET interface. PPP authentication is disabled by default. Note that authentication must be disabled on both end-points of the SONET circuit.

To enable authentication on a selected interface, use the **ppp authentication** command.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
no ppp authentication
```

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# no ppp authentication
cli#
```

Related Commands

- **interface** on page 181
- **ppp authentication** on page 322
- **show interface** on page 414

no ppp chap-hostname

Removes the Challenge Handshake Authentication Protocol (CHAP) hostname from the current POS interface. Challenge Handshake Authentication Protocol (CHAP) provides secure authentication for devices that want to participate in Point-to-Point Protocol (PPP) links.

To define the CHAP hostname on the current interface, use the **ppp chap-hostname** command.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
no ppp chap-hostname <hostname>
```

Arguments

<hostname>	The hostname that you want to remove from this interface. Valid range: 0 to 255 bytes in length.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# no ppp chap-hostname "remote device"
cli#
```

Related Commands

- **interface** on page 181
- **no ppp authentication** on page 284
- **no ppp chap-password** on page 286
- **ppp authentication** on page 322
- **ppp chap-hostname** on page 323
- **ppp chap-password** on page 324
- **show interface** on page 414

no ppp chap-password

Removes the Challenge Handshake Authentication Protocol (CHAP) password from the current POS interface. Challenge Handshake Authentication Protocol (CHAP) provides authentication for devices that want to participate in Point-to-Point Protocol (PPP) links. This command sets the CHAP password on this interface to a Null value.

To define the CHAP password on the current interface, use the **ppp chap-hostname** command.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
no ppp chap-password
```

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# no ppp chap-password
cli#
```

Related Commands

- **interface** on page 181
- **no ppp authentication** on page 284
- **no ppp chap-hostname** on page 285
- **ppp authentication** on page 322
- **ppp chap-hostname** on page 323
- **ppp chap-password** on page 324
- **show interface** on page 414

no ppp ipcp-report-address

Used when the Cuda 12000 must interact with a Juniper Networks system. This command configures the current POS interface so that it does not provide its IP address during Internet Protocol Control Protocol (IPCP) negotiations.

The Juniper router will not complete PPP negotiations if the Cuda 12000 provides its IP address during IPCP negotiations. Therefore if a PPP connection is being brought up between these two systems, you must issue the **no ppp ipcp-report-address** command. Note that the Cuda 12000 reports its IP address during IPCP negotiations by default.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
no ppp ipcp-report-address
```

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# no ppp ipcp-report-address
cli#
```

Related Commands

- **interface** on page 181
- **ppp ipcp-report-address** on page 325
- **show interface** on page 414

no ppp negotiation-count

Use this command to set the maximum number of Point-to-Point Protocol (PPP) negotiation attempts allowed on a POS interface to the default of 10. Upon exceeding the default number of negotiation attempts, the PPP connection fails.

To specify the number of maximum negotiation attempts for the current interface, use the **ppp negotiation-count** command.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
no ppp negotiation-count
```

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# no ppp negotiation-count
cli#
```

Related Commands

- **interface** on page 181
- **ppp negotiation-count** on page 326
- **show interface** on page 414

no ppp pap-sent-username

This command allows you to delete the username and password that this interface sends in response to Password Authentication Protocol (PAP) challenges from a peer.

Use the **ppp pap-sent-username** command to define the PAP username/password configuration that a POS interface sends in response to authentication challenges.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
no ppp pap-sent-username
```

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# no ppp pap-sent-username
cli#
```

Related Commands

- **interface** on page 181
- **ppp pap-sent-username** on page 327
- **show interface** on page 414

no ppp timeout

Use this command to revert back to the default PPP authentication and retry timeout values; both have a default value of 3.

Use the **ppp timeout** command to configure new PPP authentication and retry timeout values.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
no ppp timeout {authentication | retry}
```

Arguments

authentication	Use this keyword to revert back to the default authentication timeout value. This configures the interface to wait a maximum of 3 seconds for an authentication response. This parameter is currently not supported.
retry	Use this keyword to revert back to the default PPP negotiation timeout value. This configures the interface to wait a maximum of 3 seconds between PPP negotiations.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# no ppp timeout retry
cli#
```

Related Commands

- **interface** on page 181

no ppp username

Use this command to remove a username/password authentication entry from the PPP LCP Users Table on the current POS interface. After removing a username of a remote peer with this command, the POS interface will no longer authenticate the peer if PAP or CHAP authentication is required.

Use the **show interface** command to display the accounts contained within the users table. Use the **ppp username** command to add a username/password entry to the users table.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
no ppp username <name>
```

Arguments

<name>	You can enter up to 256 characters. Names that contain spaces must be enclosed within quotes.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# no ppp username "Remote Router 1"
cli#
```

Related Commands

- **interface** on page 181
- **ppp username** on page 328
- **show interface** on page 414

no proxy-arp

Disables proxy ARP on the current cable interface.

Proxy ARP enables modems to perform address resolution with other modems in the cable plant. Use this command to disable proxy ARP and prevent modems from performing address resolution with other modems in the cable plant.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
no proxy-arp
```

Example

The following example disables proxy ARP on cable interface 1/1/1:

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# show proxy-arp
Simple Proxy Arp:                    Enabled

cli:172.16.19.10:interface:cable:csi(1/1/1)# no proxy-arp

cli:172.16.19.10:interface:cable:csi(1/1/1)# show proxy-arp
Simple Proxy Arp:                    Disable

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **proxy-arp** on page 343
- **show proxy-arp** on page 508

no snmp-server contact

Removes the contact information (sysContact) for the device.

Mode

`root`

Syntax

`no snmp-server contact`

Example

```
cli:192.168.220.230:root# no snmp-server contact
cli:192.168.220.230:root#
```

Related Commands

- **snmp-server contact** on page 558

no snmp-server context

Removes an SNMP context.

Mode

root

Syntax

no snmp-server context <context-name>

Arguments

<context-name>	Name of the context that you want to remove. The name was assigned by the **snmp-server context** command.

Example

```
cli:192.168.208.3:root# show snmp context

row count: 2

Name                            Storage      Status
------------------------------- ------------ ------------
adc                             NonVolatile  Active
adcw                            NonVolatile  Active

cli:192.168.208.3:root# no snmp-server context adcw
cli:192.168.208.3:root# show snmp context

row count: 1

Name                            Storage      Status
------------------------------- ------------ ------------
adc                             NonVolatile  Active
```

Related Commands

- **show snmp context** on page 525
- **snmp-server context** on page 559

no snmp-server community

Deletes an SNMP community.

Mode

`root`

Syntax

`no snmp-server community` <community name>

Arguments

<community-name>	Name that identifies the SNMP community that you want to remove.

Example

```
cli:192.168.208.3:root# no snmp-server community beta
cli:192.168.208.3:root# show snmp community

row count: 4

Name         Security Name         Context                 Storage
-----------  ------------------    --------------------    ------------
bat          ball                                          NonVolatile
guitraps     guitraps                                      NonVolatile
private      adc                    adc                    NonVolatile
public       adc                    adc                    NonVolatile

cli:192.168.208.3:root#
```

Related Commands

- **show snmp community** on page 523
- **snmp-server community** on page 556

no snmp-server group

Deletes an SNMP security group.

Mode

root

Syntax

no snmp-server group <group-name>

Arguments

<group-name>	Name of the group assigned by the **snmp-server group** command.

Example

```
cli:192.168.220.230:root# no snmp-server group team
cli:192.168.220.230:root# show snmp group

row count: 7

Group      Context Model Level  Read View   Write View  Notify View Storage
---------- ------- ----- ------ ---------- ---------- ---------- ---------
adc        adc     V1    NoAuth public     private    public     NonVolatile
adc        adc     V2c   NoAuth public     private    public     NonVolatile
adc        adc     V3    NoAuth public     private    public     NonVolatile
bbb                V1    NoAuth v1default                       NonVolatile
bbb                V2c   NoAuth v1default                       NonVolatile
guitraps           V1    NoAuth                       guitraps   NonVolatile
guitraps           V2c   NoAuth                       guitraps   NonVolatile

cli:192.168.220.230:root#
```

Related Commands

- **show snmp group** on page 527
- **snmp-server group** on page 561

no snmp-server host

Allows you to:

- Remove an SNMP host entirely from the SNMP configuration. To do this, issue the **no snmp-server host** command with only the <ip-address> argument.
- Remove an SNMP host from the list of hosts that receive traps from the Cuda 12000.
- Remove an SNMP host from the list of hosts that receive informs from the Cuda 12000.
- Remove the mask from an SNMP host entry in the SNMP configuration.



The 127.0.0.1:54321 host entry is a special entry that CudaView uses to receive traps from the chassis. Do not delete this entry.

Mode

`root`

Syntax

```
no snmp-server host <ip-address> [{traps | informs}] [mask]
  [notification-type <type>...]
```

Arguments

<ip-address>	IP address of the SNMP host.
{**traps | informs**}	Specify one of these values: ■ traps – The specified host no longer receives traps from the Cuda 12000. ■ informs – The specified host no longer receives informs from the Cuda 12000.
mask	Removes the IP address mask from the host entry. By removing the mask from the SNMP host, you remove the host from the range of allowed hosts for an SNMP community that has been configured with the specified mask. You do not remove the host entry itself.
notification-type <type>...	Specify one or more types of notification sent to the host that you want to remove. See Table 2-3 on page 566 for more information on notification types.

Example

In this example, the user terminates sending of cold-start notifications to the host and then removes the entry for host 133.10.1.1:

```
cli:root# no snmp-server host 133.10.1.1 notification-type cold-start
cli:root# no snmp-server host 133.10.1.1
```

Related Commands

- **show snmp host** on page 528
- **snmp-server host** on page 564

no snmp-server location

Removes the description (sysLocation) of the device's location.

Mode

```
root
```

Syntax

```
no snmp-server location
```

Example

```
cli:192.168.208.3:root# no snmp-server location
cli:192.168.208.3:root#
```

Related Commands

- **snmp-server location** on page 572

no snmp-server name

Removes the system name (sysName) of the device.

Mode

root

Syntax

no snmp-server name

Example

```
cli:192.168.208.3:root# no snmp-server name
cli:192.168.208.3:root#
```

Related Commands

- **snmp-server name** on page 573

no snmp-server user

Deletes an SNMP user account.

Mode

`root`

Syntax

`no snmp-server user` <user>

Arguments

<user>	Name of the user.

Example

```
cli:192.168.220.207:root# show snmp user

row count: 3

Name                            Authentication Privacy Storage     Status
------------------------------- -------------- ------- ----------- ----------
adc                             HMAC-MD5-96    CBC-DES NonVolatile Active
template                        HMAC-MD5-96    CBC-DES Permanent   Active
dave_jones                      HMAC-MD5-96    None    NonVolatile Active

cli:192.168.220.207:root# no snmp-server user dave_jones
```

Related Commands

- **show snmp user** on page 532
- **snmp-server user** on page 574

no snmp-server view

Removes an SNMP access view.

Mode

root

Syntax

no snmp-server view <view-name>

Arguments

<view-name>	Name of the SNMP view as assigned by the **snmp-server view** command.

Example

```
cli:192.168.220.206:root# show snmp view

row count: 4

View Name        Subtree                     Type     Storage     Status
---------------- --------------------------- -------- ----------- -----------
public           1.3.6.1                     Included NonVolatile Active
private          1.3.6.1                     Included NonVolatile Active
guitraps         1.3.6.1                     Included NonVolatile Active
mgmt_view        1.3.6.2                     Included NonVolatile Active

cli:192.168.220.206:root# no snmp-server view mgmt_view
cli:192.168.220.206:root#
```

Related Commands

- **show snmp view** on page 533
- **snmp-server view** on page 576

ospf-vi

Configures OSPF virtual interfaces. OSPF requires that all areas be attached to the OSPF backbone area (area 0.0.0.0). However, you may encounter situations in which you cannot connect an OSPF area directly to the backbone. If your Cuda 12000 is an area border router between one area that is physically connected to the OSPF backbone and one area that is not (the non-contiguous area), you can create a virtual interface on your Cuda 12000 to connect the non-contiguous area to the OSPF backbone.

Before you configure an OSPF virtual interface, make sure that you have configured the OSPF backbone area (area 0).

The no form of the command, when specified with no arguments, removes the virtual interface. When specified with the authentication argument, the no form of the command disables authentication on the interface, regardless of authentication type (MD5 or simple password).

Mode

```
router:ospf
```

Syntax

```
ospf-vi <transit-area-id> <neighbor-router-id>
[{enable | disable}] [dead-interval <seconds>] [hello-interval
<seconds>] [retransmit-interval <seconds>] [transit-delay <seconds>]
[authentication {{md5 | password} | {key-id <id> key <key> | key
<key>}}]

no ospf-vi <transit-area-id> <neighbor-router-id> [authentication]
```

Arguments

<transit-area-id>	Specifies the area ID of the transit area, which is the area that connects the non-contiguous area to the backbone.
<neighbor-router-id>	Specifies the router ID of the OSPF neighbor that acts as the Area Border Router between the transit area and the backbone.
{enable | disable}	Enables or disables the virtual interface.
dead-interval <seconds>	Specifies the dead interval, which is the interval, in seconds, during which hello packets must not be seen before neighbors declare the router down. You must set this value the same for all participating OSPF routers on the virtual interface. Valid range: 0 to 2147483647 seconds. The default is 40.

hello-interval <seconds>	Specifies the interval, in seconds, the router waits between sending Hello packets on the current OSPF virtual interface. Valid range: 1 to 65535 seconds. The default is 10.
retransmit-interval <seconds>	Specifies the Link State Advertisement (LSA) retransmit interval in seconds for the current interface. This interval determines the number of seconds before a link state advertisement is retransmitted. Valid range: 1 to 65535. The default is 5.
transit-delay <seconds>	Specifies the number of seconds it takes to transmit a link state update packet on the current OSPF interface. Valid range: 1 to 65535. The default is 1.
authentication [{{**md5 | password**} **|** {**key-id** <id> **key** <key> **| key** <key>}}]	Specifies the type of authentication (MD5 or simple password) that OSPF uses as a security measure so that the current interface exchanges routing information with authorized neighbors only. This command also specifies the security key that is used for routing information exchanges. Specify one of these authentication types: md5 – Specifies that the OSPF virtual interface uses MD5 authentication. If you specify md5 as the authentication type, you must subsequently specify the shared secret key with the key-id <id> key <key> argument. See the description of the ip ospf authentication command for more information on MD5. password – Specifies a simple password authentication method that includes a password in all OSPF messages. If you specify password as the authentication type, you must subsequently specify the password with the key <key> argument. See the description of the ip ospf authentication command for more information on simple password authentication. Specify one of these secret keys: key-id <id> key <key> – Specifies the md5 key ID and shared secret key. The key-id <id> parameter is an integer ranging from 1 to 255 that identifies the shared secret key. The key <key> parameter specifies the key, which is a text string from 1 to 16 characters. key <key> – Specifies the simple password, which is a text string from 1 to 16 characters.

Example

```
cli:# router ospf
mode: router:ospf
cli:# ospf-vi 0.0.0.2 133.132.2.2
cli:# ospf-vi 0.0.0.2 133.132.2.2 dead-interval 55
cli:# ospf-vi 0.0.0.2 133.132.2.2 hello-interval 20
cli:# ospf-vi 0.0.0.2 133.132.2.2 retransmit-interval 15
cli:# ospf-vi 0.0.0.2 133.132.2.2 transit-delay 3
cli:# ospf-vi 0.0.0.2 133.132.2.2 authentication md5
cli:# ospf-vi 0.0.0.2 133.132.2.2 authentication key-id 32459 key
my-secret-key
cli:# show ospf interface virtual
row count: 1

Transit       Neighbor    State Transit Retrans Hello Dead Auth Event Status
Area                            Delay   Int     Int   Int  Type Count
------------- ----------- ----- ------- ------- ----- ---- ---- ----- -----
0.0.0.2       133.132.2.2 down        3      15    20   55 MD5      0 Active
```

Related Commands

- **show ospf** on page 489

ospf

Use this command within **router** mode to enter OSPF configuration mode.

Mode

```
router
```

Syntax

```
ospf
```

Example

```
cli:172.16.19.10:root# router
mode: router
cli:172.16.19.10:router# ospf
mode: router:ospf
cli:172.16.19.10:router:ospf# show mode
mode: router:ospf
cli:172.16.19.10:router:ospf#
```

Related Commands

- **router** on page 356

ospf area

Use this command to configure an OSPF area.

Mode

```
router:ospf
```

Syntax

```
ospf area <area id>
[authentication {md5 | password}]
[[stub [no-summary]]
[default-cost <cost>]
[range <ip-address> <mask> [advertise-matching]]
[{enable | disable}]
```

Arguments

<area id>	Identifier of the area that you want to configure, specified in the IP address form.
authentication {**md5 | password**}	Specify one of these authentication types for the area: md5 – Specifies that the area uses MD5 authentication. See the description of the **ip ospf authentication** command for more information on MD5. password – Specifies a simple password authentication method for the area that includes a password in all OSPF messages. See the description of the **ip ospf authentication** command for more information on simple password authentication.
stub	Defines the area as a stub area.
range <ip address> <mask> [**advertise-matching**]	Specifies an area range using an IP address/mask combination. Use an area range to consolidate and summarize routes. The optional **advertise-matching** argument specifies that the Cuda 12000 will advertise the IP address and mask to other routers. Otherwise, the IP address and mask are not advertised.

default-cost	Defines the default metric for this area. Enter a value between 1 and 65535 (the default value is 1). This value specifies the type of service (TOS) cost. The lowest TOS cost has the highest priority for routing. For example, if two interfaces can be used to forward traffic to the same destination, the one with the lower TOS cost is selected. The **ip ospf cost** command allows you to override the default cost on each OSPF interface. See the description of the **ip ospf cost** command for more information.
enable	Enables the specified OSPF area.
disable	Disables the specified OSPF area.

Example

```
cli# router ospf
mode: router:ospf
cli# ospf area 0.0.0.1 authentication md5
cli# ospf area 0.0.0.1 stub
cli# ospf area 0.0.0.1 default-cost 10
cli#
```

Related Commands

- **ip ospf area-id** on page 191
- **ip ospf authentication** on page 193
- **ip ospf cost** on page 196
- **no ospf area** on page 278
- **show ospf** on page 489

override

Use this command to set the override rules for import or export route map filters. These rules allow the administrator to change route data as packets are received or transmitted. Use this command while in route-map configuration mode.

Mode

To set override rules for RIP import filters:

```
router:rip:import:route-map(#)
```

To set override rules for RIP export filters:

```
router:rip:export:route-map(#)
```

To set override rules for OSPF import filters:

```
router:ospf:import:route-map(#)
```

To set override rules for OSPF export filters:

```
router:ospf:export:route-map(#)
```

Syntax

Within router:rip:import:route-map (#) mode:

```
override {metric <metric-value> | tag <tag-value> |
preference <preference-value>}
```

Within router:rip:export:route-map(#)mode:

```
override {metric <metric-value> | tag <tag-value>}
```

Within router:ospf:import:route-map(#)mode:

```
override preference <preference-value>
```

Within router:ospf:export:route-map(#):

```
override {metric <metric-value> | tag <tag-value>}
```

Arguments

<metric-value>	Overrides the cost metric associated with matching routes.
<tag-value>	Overrides the tag value associated with matching routes.
<preference-value>	Overrides the preference associated with matching routes.

Example

```
cli# router ospf export
mode: router:ospf:export
cli# route-map 1 deny
cli# show mode
mode: router:ospf:export:route-map(1)
cli# override metric 20
cli# override tag 50
cli# show route-map 1
ID                                      1
Description
Route Address                   0.0.0.0
Route Mask                      0.0.0.0
Type                               NONE
Specific1                       0.0.0.0
Specific2                       0.0.0.0
Tag                                   0
Key Bits                              0
Metric                               20
Flags                                80
Action Tag                           50
```

Related Commands

- **map-list** on page 251
- **match** on page 254
- **route-map** on page 354
- **show route-map** on page 517

periodic-ranging-interval

Specifies how often this cable interface periodically invites modems to range.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
periodic-ranging-interval <value>
```

Arguments

<value>	Interval, in seconds, between ranging invitations. Acceptable values: 5–30. Default: 15 seconds.

Example

```
cli# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli# periodic-ranging-interval 20
cli#
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421

ping

Use this command to send Internet Control Message Protocol (ICMP) echo request packets to a node on your cable network to see if it is reachable and online.

Mode

Any.

Syntax

```
ping [-l <size>] [-n <count>] [-w <timeout>]
[-s <src-ip-address>] <dest-ip-address>
```

Arguments

-l <size>	Size of the ping to send. Valid range: 64 – 64000 bytes. If you do not specify a size, the size of the ping defaults to 64 bytes.
-n <count>	Number of pings to send. Valid range: 0 – 1000. A value of 0 means forever. If you do not specify the number of pings, then 5 pings are sent by default.
-w <timeout>	Number of seconds to wait for each reply. Valid range: 1 – 30. If you do not specify a timeout, then the ping command uses the default timeout (2 seconds).
-s <src-ip-address>	Specifies the IP address (which you must supply as an IP number, not a hostname) on your Cuda 12000 that will be the source address in outgoing echo request packets. By default, the source address is the IP address of the interface on which the Cuda 12000 sends the echo request packet. If your Cuda 12000 has more than one IP address, this argument enables you to override the default source address.
<dest-ip-address>	IP address of destination host that you want to ping.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem

row count: 12

MAC Address       IP Address      SID  CID  CPE D:U Power  Timing Modem
                                                   (dbMV)        State
----------------- --------------- ---- ---- --- --- ------ ------ ----------
00:90:96:00:29:71 201.1.1.112       1    1   0 1:2      0   2216 Registered
00:90:83:36:82:ee 201.1.1.108       2    0   0 1:2      0   1248 RegBpiTek
00:90:96:00:29:6d 201.1.1.104       3    1   0 1:2      0   2217 Registered
00:10:95:04:0a:b7 201.1.1.109       4    1   0 1:2      0   2725 Registered
00:90:96:00:39:f9 201.1.1.101       5    1   0 1:2      0   2212 Registered
00:10:95:01:ef:d8 201.1.1.100       6    1   0 1:2      0   9647 Registered
00:10:95:04:0a:c3 0.0.0.0           7   57   0 1:2     -9   2725 Ranging
00:90:96:00:39:7f 201.1.1.102       8    1   0 1:2      0   2219 Registered
00:10:95:04:0a:c4 201.1.1.110       9    1   0 1:2      0   2724 Registered
00:10:95:01:f0:05 201.1.1.103      10    1   0 1:2      0  26266 Registered
00:90:83:32:9f:8c 201.1.1.106      11    1   0 1:2      0   1639 Registered
00:90:83:36:82:f1 201.1.1.107      12    1   0 1:2      0   1239 Registered

cli:192.168.208.3:interface:cable:csi(1/1/1)# ping -n 3 -w 8 201.1.1.112
Sending 3, 64-byte ICMP Echos to 201.1.1.112, timeout is 8 seconds
!!!
Success rate is 100.0 percent (3/3)
```

Related Commands

- **traceroute** on page 593

plant-delay

Specifies the estimated plant propagation delay, in microseconds.

Mode

```
interface:cable:csi(c/s/p)
```

Syntax

```
plant-delay <value>
```

Arguments

<value>	Propagation delay, in microseconds. Valid range: 400-1600 microseconds. Default: 400 microseconds.

Example

```
cli# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli# plant-delay 500
cli#
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421
- **show mac** on page 454

pll-state

Sets the phase lock loop state for the current cable interface.



CAUTION: For use by an expert-level administrator only. Configuring this option may impact CMTS operation.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
pll-state
```

Example

```
cli# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli# pll-state
cli#
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421
- **show mac** on page 454
- **show pll-state** on page 492

pos flag

Use this command to set values for the following SONET frame overhead bytes:

- **c2** — This byte serves as a signal label and indicates the content held within the synchronous payload envelope (SPE) of the SONET frame, as well as the status of that payload.
- **J0** — The section trace byte required for interoperability with some SDH devices in Japan.

Use the **no pos-flag** command to clear these overhead bytes.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
pos flag {c2 | j0} <value>
```

Arguments

c2 <value>	Path signal identifier used to identify the payload content type. Valid range: 0 – 255. Default:0xcf.
j0 <value>	Section trace byte. For interoperability with SDH equipment. Valid range: 0 – 255. Default:0xcc.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# pos flag c2 250
cli#
```

Related Commands

- **show interface pos** on page 437

pos report

Configures the POS interface to report selected SONET alarms to BASMONITOR. The plop, sf-ber, slof, and slos SONET alarms are reported by default.

Use the **no pos report** command to disable logging of select SONET alarms.

Alarm reporting follows the SONET alarm hierarchy rules. Only the most severe alarm of an alarm group is reported.

The following table lists the SONET Near-End Failure Alarm hierarchy:

Table 2-1 SONET Near-End Failure Alarm Hierarchy

Priority	Near-End Failures
Highest	LOS
	LOF
	LAIS
	PAIS
	PLOP

The following table lists the SONET Far-End Failure Alarm hierarchy:

Table 2-2 SONET Far-End Failure Alarm Hierarchy

Priority	Near-End Failures
Highest	LRDI
Lowest	PRDI

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
pos report {lais|lrdi|pais|plop|prdi|sd-ber|sf-ber|slof|slos}
```

Arguments

lais	Configures the POS interface to report line alarm indication signal errors.
lrdi	Configures the POS interface to report line remote defect indication errors.
pais	Configures the POS interface to report path alarm indication signal errors. Used to alert the downstream path terminating equipment (PTE) that the interface has detected a defect on its incoming line signal.
plop	Configures the POS interface to report path loss of pointer errors. Reported by default. This defect is reported as a result of either of the following: ■ Invalid pointer (H1, H2) ■ Excess number of new data flag (NDF) enabled indications.
prdi	Configures the POS interface to report path remote defect indication errors.
sd-ber	Configures the POS interface to report signal degradation BER errors.
sf-ber	Configures the POS interface to report signal failure BER errors. Reported by default.
slof	Configures the POS interface to report section loss of frame errors. Reported by default. Detected when the POS interface sees a severely error framing (SEF) defect on the incoming SONET signal for 3 milliseconds.
slos	Configures the POS interface to report section loss of signal errors. Reported by default. Detected when the POS interface sees an all-zeros pattern on the incoming SONET signal which lasts 19 microseconds or longer.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# pos report lais
cli#
```

Related Commands

- **no pos report** on page 281
- **pos flag** on page 316
- **pos scramble** on page 320
- **pos threshold** on page 321

pos scramble

Configures scrambling of the synchronous payload envelope (SPE) on the current POS interface. Scrambling ensures sufficient bit transition density of the SPE. Note that scrambling must be enabled on both end-points of a POS connection. SPE payload scrambling is disabled by default.

Use the **no pos scramble** command to disable scrambling on the current POS interface.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
pos scramble
```

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# pos scramble
cli#
```

Related Commands

- **no pos scramble** on page 282
- **show interface pos** on page 437

pos threshold

Use this command to set the Signal degrade BER (sd-BER) and Signal failure BER (sf-BER) alarm Bit Error Rate (BER) threshold values. Note that you enable reporting of sd-ber and sf-ber defects using the **pos-report** command.

When the signal degrade or signal failure bit error rate meets or exceeds the corresponding threshold, an alarm is sent to the POS controller. To view the current BER threshold settings, as well as the defects that have been reported, use the **show controllers** command.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
pos threshold {sd-ber | sf-ber} <rate>
```

Arguments

sd-ber	Sets the threshold for signal degrade BER errors on the current interface. The default is 6 (10 -6).
sf-ber	Sets the threshold for signal failure BER errors on this interface. Note that sf-ber alarms are reported by default. The default is 3 (10 -3).
<rate>	Specifies the threshold.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# pos threshold sd-ber 7
cli# pos threshold sf-ber 4
```

Related Commands

- **no pos threshold** on page 283
- **pos report** on page 317
- **show controllers** on page 392

ppp authentication

Enable authentication on the current POS interface and specify the order in which PAP and CHAP authentication protocols are used. Use this command to enable Challenge Handshake Authentication Protocol (CHAP), Password Authentication Protocol (PAP), or both.

Note that this command specifies only which protocols — PAP, CHAP, or both — the POS interface uses to authenticate remote devices. You must also define usernames and passwords associated with both PAP and CHAP authentication using the **ppp username** command.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
ppp authentication {chap | chap pap | pap chap | pap}
```

Arguments

chap	Enables CHAP on the current POS interface.
chap pap	Enables negotiation of CHAP first, then tries PAP.
pap chap	Enables negotiation of PAP first, then tries CHAP.
pap	Enables PAP on the current POS interface.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# ppp authentication chap
```

Related Commands

- **no ppp authentication** on page 284
- **ppp chap-hostname** on page 323
- **ppp chap-password** on page 324
- **ppp pap-sent-username** on page 327
- **ppp username** on page 328
- **show interface** on page 414

ppp chap-hostname

Configures the hostname that this POS interface uses for Challenge Handshake Authentication Protocol (CHAP) authentication. From a CHAP server perspective, you must configure the CHAP hostname using this command.

When configuring the interface as a CHAP client, you must use this command to configure the chap-hostname with which the interface will respond to challenges, and, in addition, use the **ppp chap-password** command to configure the associated password.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

ppp chap-hostname <name>

Arguments

<name>	Username that this interface uses for CHAP authentication. You can enter up to 256 characters.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# ppp authentication chap
cli# ppp chap-hostname "Trusted Router"
cli#
```

Related Commands

- **no ppp chap-password** on page 286
- **ppp chap-password** on page 324
- **show interface pos** on page 437

ppp chap-password

Configures the password that this interface sends in response to CHAP challenges from remote peers.

CHAP provides authentication for devices that want to establish a Point-to-Point Protocol (PPP) link with the current POS interface. The CHAP password, along with the CHAP hostname, must be provided by remote devices during CHAP challenges before a PPP connection is established with the interface.

Note that you must set the CHAP hostname using the **ppp chap-hostname** command before you configure a CHAP password. Use the **no ppp chap-password** command to delete the CHAP password from the current interface.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
ppp chap-password <password>
```

Arguments

<password>	CHAP password that you want to define for this interface. Note that a remote router must provide the correct CHAP hostname and password before establishing a PPP link with the interface. The password can be 0 to 255 bytes in length.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# ppp chap-password p210
cli#
```

Related Commands

- **no ppp chap-password** on page 286
- **no ppp chap-hostname** on page 285

ppp ipcp-report-address

Configures the current POS interface to provide its IP address during Internet Protocol Control Protocol (IPCP) negotiations when creating a point-to-point link.

When creating a link with a Juniper Networks router, you must disable IPCP address reporting using the **no ppp ipcp-report-address** command; the Juniper router will not complete PPP negotiations if the Cuda 12000 provides its IP address during IPCP negotiations.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
ppp ipcp-report-address
```

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# ppp ipcp-report-address
cli#
```

Related Commands

- **no ppp ipcp-report-address** on page 287
- **show interface pos** on page 437

ppp negotiation-count

Specifies the maximum number of Point-to-Point (PPP) negotiation attempts the POS interface allows while attempting to establish a PPP connection. Upon exceeding the number of negotiation attempts with this command, the PPP connection fails. By default, the maximum number of PPP negotiation attempts allowed on a POS interface is 10.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
ppp negotiation-count <value>
```

Arguments

<value>	Maximum number of PPP negotiation attempts allowed for this interface when attempting to set up a PPP connection. Valid range: 0 to 100. The default is 10.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# ppp negotiation-count 20
cli#
```

Related Commands

- **no ppp negotiation-count** on page 288

ppp pap-sent-username

Configures the username and password that the interface sends in response to Password Authentication Protocol (PAP) requests.

Similar to Challenge Handshake Authentication Protocol (CHAP), PAP provides password authentication of remote routers that attempt to establish a point-to-point connection with a POS interface. If the incorrect username and password is provided, the connection is not allowed.

Although both PAP and CHAP are authentication mechanisms used with point-to-point protocol (PPP), PAP is less secure than CHAP. PAP does not encrypt the password; CHAP does.

Use the **no ppp pap-sent-username** command to remove a PAP username/password configuration from the POS interface.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
ppp pap-sent-username <username> password <password>
```

Arguments

<username>	PAP username that you want to define for the POS interface. The interface will respond to PAP requests with this username. This name may range from 1 to 255 characters. Passwords that contain spaces must be enclosed within quotes.
<password>	PAP password with which the interface will respond to PAP requests. The password may range from 1 to 255 uppercase and lowercase alphanumeric characters. Usernames that contain spaces must be enclosed within quotes.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# ppp pap-sent-username "router 101" password 12stk123
```

Related Commands

- **no ppp pap-sent-username** on page 289

ppp username

This command allows you to define user authentication accounts on the current POS interface. Each username/password combination is added to the PPP LCP Users Table for that interface. These username and password pairs are used for any authentication requests received from both CHAP and PAP.

You can define multiple username/password pairs. Note that you must define a username/password entry for each remote system from which the current POS interface requires authentication.

Use the **no ppp username** command to delete a username/password entry from the current POS interface.

Mode

```
interface:pos:csi(<c/s/i>)
```

Syntax

```
ppp username <username> password <password>
```

Arguments

<username>	Username that identifies the remote system during PAP and CHAP negotiations. Username may range from 1 to 255 characters.
<password>	Password that the remote system must provide in negotiations with the POS interface in order to establish a point-to-point connection. Password may range from 1 to 255 uppercase and lowercase alphanumeric characters.

Example

```
cli# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli# ppp username "Router 1" password Chicago_and_LA
cli#
```

Related Commands

- **no ppp username** on page 291
- **show interface** on page 414

privacy auth

Specifies the lifetime, in seconds, that the CMTS assigns to an authorization key for a specified cable modem.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
privacy auth <mac-address> {cm-lifetime <number> | cm-reset
{invalidateAuth | invalidateTeks | sendAuthInvalid}}
```

Arguments

<mac-address>	Specifies the MAC address of the cable modem to which the CMTS assigns an authorization key.
{**cm-lifetime** <number> **|**	Specifies the lifetime in seconds that the CMTS assigns to an authorization key for this cable modem. Values range from 1 to 6048000.
cm-reset {**invalidateAuth | invalidateTeks | sendAuthInvalid**}	Specifies one of the following values: ■ invalidateAuth – The CMTS invalidates the current cable modem authorization keys, but does not transmit an authorization message or invalidate unicast TEKs. ■ invalidateTeks – The CMTS invalidates the current authorization key and transmits an authorization invalid message to the cable modem. The CMTS also invalidates all unicast TEKs associated with this cable modem authorization. ■ sendAuthInvalid – The CMTS invalidates the current cable modem authorization key and transmits an invalid message to the cable modem. The CMTS does not invalidate the unicast TEKs.

Example

```
cli:interface:cable:csi(1/1/1)# show modem

row count: 11

MAC Address       IP Address      SID  CID  CPE D:U Power  Timing Modem
                                                   (dbMV)        State
----------------- --------------- ---- ---- --- --- ------ ------ ----------
00:10:95:04:0a:c4 201.1.1.107      479    1   0 1:2      0   2725 Registered
00:10:95:04:0a:bd 201.1.1.108      502    1   0 1:2      0   2726 Registered
00:90:96:00:29:6d 201.1.1.101      495    1   0 1:2      0   2218 Registered
00:90:96:00:39:f9 201.1.1.105      499    1   0 1:2      0   2215 Registered
00:10:95:04:0a:b7 201.1.1.102      475    1   0 1:2      0   2722 Registered
00:10:95:04:0a:c3 0.0.0.0            6    0   0 1:2     -9   2727 Ranging
00:10:95:01:f0:05 201.1.1.104      491    1   0 1:2      0   2206 Registered
00:10:95:01:ef:d8 201.1.1.103      492    1   0 1:2      0   2208 Registered
00:90:96:00:29:71 201.1.1.106      497    1   0 1:2      0   2215 Registered
00:90:83:32:9f:8c 201.1.1.110      507    1   0 1:2      0   1638 Registered
00:90:83:36:82:f1 201.1.1.111      510    1   0 1:2      0   1233 Registered
cli:interface:cable:csi(1/1/1)# privacy auth 00:90:83:32:9f:8c cm-lifetime
604800
```

Related Commands

- **show privacy auth** on page 493

privacy base auth-lifetime

Specifies the default lifetime, in seconds, that the CMTS assigns to a new authorization key.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
privacy base auth-lifetime <number>
```

Arguments

<number>	Specifies the allowed value range that a CMTS can assign to a new authorization key. The default value is 6048000. Values range from 1 to 6048000.

Example

This example shows how to set the base authorization lifetime in seconds:

```
cli:interface:cable:csi(1/1/1)# privacy base auth-lifetime 43000
cli:interface:cable:csi(1/1/1)#
```

Related Commands

- **show privacy base** on page 495

privacy base cert-trust

Specifies the default level of trust for all new self-assigned manufacturer certificates. This command applies to BPI+ only.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
privacy base cert-trust {trusted | untrusted}
```

Arguments

trusted	Specifies that all new certificates are trusted. Trusted certificates are valid certificates.
untrusted	Specifies that all new certificates are untrusted. Untrusted certificates are invalid certificates.

Example

This example shows how to set the trust of new self-assigned certificates:

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# privacy base cert-trust trusted
cli:192.168.208.3:interface:cable:csi(1/1/1)#
```

Related Commands

- **privacy ca-cert** on page 335
- **privacy cm-cert** on page 337
- **show privacy base** on page 495

privacy base enable-cert-validity-periods

Specifies whether certificates have their validity period checked against the current time of day. This command applies to BPI+ only.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
privacy base enable-cert-validity-periods {true | false}
```

Arguments

{**true** | **false**} Specifies one of the following values:

- true – Sets the certificates to true. This means that their validity is checked against the current time of day.
- false – Sets the certificates to false. This means that their validity is not checked against the current time of day.

Example

```
cli:cable:csi(1/1/1)# privacy base enable-cert-validity-periods true
```

Related Commands

- **show privacy auth** on page 493
- **show privacy base** on page 495
- **show privacy cm-cert** on page 500

privacy base tek-lifetime

Specifies the default lifetime, in seconds, that the CMTS assigns to a new Traffic Encryption Key (TEK).

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

privacy base tek-lifetime <number>

Arguments

<number>	Specifies the lifetime, in seconds, that the CMTS assigns to new TEKs. The default is 43200. Values range from 1 to 604800.

Example

This example shows how to set the base TEK lifetime in seconds:

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# privacy base tek-lifetime 50000
cli:192.168.208.3:interface:cable:csi(1/1/1)#
```

Related Commands

- **show privacy tek** on page 506

privacy ca-cert

Specifies manufacturer certification authority (CA) X.509 certificates. The manufacturer CA certificate is used to validate the authenticity of manufacturer CAs who issue cable modem (CM) certificates. This command applies to BPI+ only.

When you specify a manufacturer CA certificate, you assign it an index number and a level of trust. You must also specify the filename of the certificate.

The **no privacy ca-cert** <number> command removes the manufacturer CA certificate.

Mode

```
root
interface:cable:csi(<c/s/i>)
```

Syntax

```
privacy ca-cert <number> [{trusted | untrusted | chained | root}]
  certificate <filename>

no privacy ca-cert <number>
```

Arguments

<number>	Specifies an index number for the manufacturer CA certificate. Values range from 1 to 10000.
[{**trusted** | **untrusted** | **chained** | **root**}]	Specifies one of the following levels of trust (see the *DOCSIS Baseline Privacy Plus Interface Specification* for more information on levels of trust): ■ trusted – Specifies that the certificate is trusted. Trusted certificates are valid certificates. ■ untrusted – Specifies that the certificate is untrusted. Untrusted certificates are invalid certificates. ■ chained – Specifies that the certificate's level of trust is chained. In order for a chained certificate to be valid, it must meet several criteria, such as: * The certificate is linked to a Root, Trusted, or Valid certificate * The certificate's signature can be verified with the issuer's public key * The current time falls within the validity period of each Chained or Root certificate within the certificate chain ■ root – Specifies that the certificate's level of trust is root. Note that only the DOCSIS Root CA Certificate (a self-signed certificate containing the DOCSIS Root CA's trusted public key) must be marked as Root. However, a CMTS MAY support multiple Root CA Certificates. At least one root certificate must be provisioned.
certificate <filename>	Specifies the filename of the manufacturer CA certificate. These certificates reside in **/bas/data/certification**.

Example

```
cli:192.168.208.3:root# privacy ca-cert 5 trusted certificate manf64.cer
```

Related Commands

- **show privacy ca-cert** on page 497

privacy cm-cert

Assigns an X.509 CM certificate to a cable modem. These certificates are issued by cable modem manufacturers. This command applies to BPI+ only.

When you assign a CM certificate to a cable modem, you assign it a level of trust. You must also specify the filename of the certificate.

The **no privacy cm-cert** <mac> command removes the specified cable modem's certificate.

Mode

```
root
interface:cable:csi(<c/s/i>)
```

Syntax

```
privacy cm-cert <mac> [{trusted | untrusted}] certificate <filename>

no privacy cm-cert <mac>
```

Arguments

<mac>	Specifies the MAC address of the cable modem.
[{**trusted** | **untrusted**}]	Specifies one of the following levels of trust: ■ trusted – Specifies that the certificate is trusted. Trusted certificates are valid certificates. ■ untrusted – Specifies that the certificate is untrusted. Untrusted certificates are invalid certificates. See the *DOCSIS Baseline Privacy Plus Interface Specification* for more information on levels of trust.
certificate <filename>	Specifies the filename of the CM certificate. These certificates reside in **/bas/data/certification**.

Example

```
cli:root# privacy cm-cert 00:10:95:04:0a:c3 trusted certificate cm64.cer
```

Related Commands

- **show privacy cm-cert** on page 500

privacy encryption

Specifies the type of encryption used for baseline privacy on the current cable interface. Both 40-bit and 56-bit data encryption standard (DES) encryption algorithms are supported. This encryption is used for transmitting keys.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
privacy encryption {40-bit-des | 56-bit-des}
```

Arguments

40-bit-des	Configures the interface for 40-bit baseline privacy encryption.
56-bit-des	Configures the interface for 56-bit baseline privacy encryption.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# privacy encryption 56-bit-des
cli:192.168.208.3:interface:cable:csi(1/1/1)#
```

Related Commands

- **show privacy base** on page 495

privacy multicast ip

Maps an IP multicast address to a security association (SA) and its associated encryption and authentication parameters. Use the no form of the command to remove the mapping. This command applies to BPI+ only.

A CMTS may map downstream multicast flows to any of three classes of BPI+ security associations: primary, static, or dynamic. If you map an IP multicast group's traffic to a primary SA, only the single cable modem that belongs to that SA can access that multicast group. If you map the multicast group to a static or dynamic SA, then multiple cable modems may access that multicast group.

Use the **privacy multicast mac** command to assign multicast SAs to specific cable modems.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
privacy multicast ip <index> <multicast-ip-address> <mask>
  said <number> sa-type {dynamic | none | primary | static}
  encrypt-alg {des40cbcMode | des56cbcMode | none} authent-alg none

no privacy multicast ip <index>
```

Arguments

<index>	Specifies an index value that identifies the IP multicast address mapping entry. Values range from 1 to 10000.
<multicast-ip-address>	Specifies the Class D IP address (for example, 239.1.1.1) of the multicast group to which you are applying the security association specified by the SAID.
<mask>	Specifies the mask, in dot-notation format, that can be used with a single multicast group address or to specify a multicast address range. For example, a multicast address of 239.1.0.0 and a mask of 255.255.0.0 means that the SA applies to all multicast groups within 239.1.0.0. For a single multicast group address (for example, 239.2.2.2), specify a mask of 255.255.255.255.
said <number>	Specifies the Security Association Identifier (SAID) of the SA. Values range from 8192 to 16383.

sa-type {**dynamic** | **none** | **primary** | **static**}	Specifies one of the following security association types: ■ dynamic – Specifies a Dynamic Security Association, which is established and eliminated on the fly in response to the initiation and termination of specific (downstream) traffic flows. Both Static and Dynamic SAs can by shared by multiple CMs. ■ none – Specifies no security association. ■ primary – Specifies Primary Security Association, which is tied to a single cable modem, and is established when that cable modem completes DOCSIS MAC registration. ■ static – Specifies a Static Security Association, which is provisioned within the CMTS.
encrypt-alg {**des40cbcMode** | **des56cbcMode** | **none**}	Specifies one of the following encryption algorithms: ■ des40cbcMode – Specifies 40-bit DES packet data encryption. ■ des56cbcMode – Specifies 56-bit DES packet data encryption. ■ none – Specifies no encryption.
authent-alg none	Specifies the authentication algorithm. At this time, **none** is the only supported value (no authentication).

Example

```
cli:interface:cable:csi(1/1/1)# privacy multicast ip 1 239.2.2.2
255.255.255.255 said 8192 sa-type dynamic encrypt-alg des56cbcMode
authent-alg none
cli:interface:cable:csi(1/1/1)# show privacy multicast ip

row count: 1

Index IP Address      IP Mask         SAID  SA Type Encrypt Alg  Authent
                                                                 Alg
----- --------------- --------------- ----- ------- ------------ -------
    1 239.2.2.2       255.255.255.255  8192 dynamic des56cbcMode    none
```

Related Commands

- **privacy multicast mac** on page 341
- **show privacy multicast** on page 504

privacy multicast mac

Assigns a multicast security association to a cable modem, thereby authorizing the modem for access to a specific downstream multicast flow. This command applies to BPI+ only.

This security association is created using the **privacy multicast ip** command. Use the **no privacy multicast mac** command to remove the security association from a cable modem.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
privacy multicast mac <mac-address> said <number>

no privacy multicast mac <mac-address> said <number>
```

Arguments

<mac-address>	Specifies the MAC address of the cable modem.
said <number>	Specifies the SAID of the multicast SA. Use the **show privacy multicast ip** command to display available SAIDs. Values range from 8192 to 16383.

Example

```
cli:cable:csi(1/1/1)# show privacy multicast ip

row count: 1

Index IP Address      IP Mask         SAID  SA Type Encrypt Alg  Authent
                                                                 Alg
----- --------------- --------------- ----- ------- ------------ -------
    1 239.2.2.2       255.255.255.255  8192 dynamic des56cbcMode    none

cli:cable:csi(1/1/1)# privacy multicast mac 00:10:95:04:0a:c3 said 8192
```

Related Commands

- **privacy multicast ip** on page 339
- **show privacy multicast** on page 504

privacy tek

Sets the lifetime, in seconds, that the CMTS assigns to a Traffic Encryption Key (TEK) for an associated SAID.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
privacy tek <said> {tek-lifetime <number> | reset}
```

Arguments

<said>	Specifies the SAID of the security association. The range of identifier values is 1 to 4294967295.
{**tek-lifetime** <number> | **reset**}	Specifies one of the following values: ■ teklifetime <number> – Specifies the allowable value range for the TEK lifetime. Values range from 1 to 6048000 seconds. ■ reset – Resets the value range for the TEK lifetime.

Example

```
cli:interface:cable:csi(1/1/1)# privacy tek 8192 tek-lifetime 604800
cli:interface:cable:csi(1/1/1)#
```

Related Commands

- **privacy base tek-lifetime** on page 334
- **show privacy base** on page 495
- **show privacy tek** on page 506

proxy-arp

Enables proxy ARP on the current cable interface. Modems cannot directly perform address resolution with other modems in the cable plant as upstream and downstream ports are separate interfaces. Enabling proxy ARP with this command allows cable modems to perform address resolution with other cable modems.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

proxy-arp

Example

The following example enables proxy ARP so that the cable modems attached to cable interface 1/1/1 can resolve addresses for other modems attached to the same interface.

```
cli# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli# proxy-arp
cli#
```

Related Commands

- **no proxy-arp** on page 292
- **show proxy-arp** on page 508

qos permission

Enables cable modem registration access to the CMTS. Use the **no qos permission** command to remove the configuration.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

qos permission modems

no qos permission modems

Arguments

modems	Enables cable modem registration requests to create entries in the QoS tables.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# qos permission modem
cli:172.16.19.10:interface:cable:csi(1/1/1)# show qos permission
permission: 4
C/S/I                                1/1/1
Permission
  Create by SNMP   : No
  Create by Modems : Yes
  Update by SNMP   : No
```

Related Commands

- **show interface cable <c/s/i> qos** on page 428

quit

Exits the Cuda 12000 command line interface (CLI) shell. Console and secure shell (SSH) users are returned to the Linux prompt. Telnet sessions are terminated.

Mode

Any.

Syntax

quit

Example

```
cli:192.168.208.3:root# quit
```

radius-server

Specifies a RADIUS authentication server. Use the no form of the command to remove the RADIUS authentication server.

Mode

root

Syntax

radius-server {**host** <ip-address> | **key** {<number> | <string>}}

no radius-server

Arguments

host <ip-address>	Specifies the IP address of the RADIUS authentication server.
key {<number> | <string>}	Specifies an encryption key as a number or string. The key is used for RADIUS authentication.

Example

```
cli:192.168.208.3:root# radius-server host 201.1.1.1
cli:192.168.208.3:root# show radius-server
radius-server host 201.1.1.1
cli:192.168.208.3:root#
```

Related Commands

- **show radius-server** on page 510

ranging-attempts

Specifies the number of times a cable modem will be invited to range before the cable modem is removed from the system. A value of zero means that the CMTS will invite the cable modem to range forever.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
ranging-attempts <value>
```

Arguments

<value>	Number of times the CMTS will invite the modem to range before removing it from the system. Valid Range: 0 – 1024. Default:16.

Example

```
cli# interface 1/1/1
cli# ranging-attempts 0
cli#
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421
- **show mac** on page 454

report

Enables the sending of the OSPF neighbor state trap and the OSPF virtual neighbor state trap. Use the no form of the command to disable sending of these traps.

Mode

```
router:ospf
```

Syntax

```
report {ospf-nbr-state | ospf-virt-nbr-state}

no report {ospf-nbr-state | ospf-virt-nbr-state}
```

Arguments

{**ospf-nbr-state** | **ospf-virt-nbr-state**}	Specifies one of the following values: ■ ospf-nbr-state – Enables sending of the OSPF neighbor state trap. When used with the no form of the command, this argument disables sending of the OSPF neighbor state trap. ■ ospf-virt-nbr-state – Enables sending of the OSPF virtual neighbor state trap. When used with the no form of the command, this argument disables sending of the OSPF virtual neighbor state trap. In addition to enabling these traps using the **report** command, you must also configure their corresponding notification types using the **snmp-server host** command in order to send the traps to SNMP hosts.

Example

```
cli:192.168.220.230:router:ospf# report ospf-nbr-state
cli:192.168.220.230:router:ospf# report ospf-virt-nbr-state
cli:192.168.220.230:router:ospf# show ospf
Admin Status                          Enabled
TOS Support                             False
Router Id                           201.1.1.1
ABR Status                              False
ASBR Status                             False

Report ospf-nbr-state                 Enabled
Report ospf-virt-nbr-state            Enabled
```

Related Commands

- **router ospf** on page 359
- **show ospf** on page 489
- **snmp-server host** on page 564

reset

Use this command to reboot a module. This command resets an active slot. When you reset the slot, the module that resides in the selected slot reboots. You can use this command to remotely reboot the management module in slot 1/13.

If you issue the command with no arguments, a soft reboot is performed.

Mode

```
slot(<c/s>)
```

Syntax

```
reset [{hard | soft}]
```

Arguments

{hard | soft}	Specifies one of the following values: ■ hard – Resets the module without rebooting it. ■ soft – Reboots the module.

Example

The following example resets the module that is installed in slot 1/1.

```
cli:172.16.19.10:root# slot 1/3
mode: slot(1/3)
cli:172.16.19.10:slot(1/3)# reset
cli:172.16.19.10:slot(1/3)#
```

Related Commands

- **slot** on page 555

reset rip stats

Sets RIP statistical counters to 0. Use this command in conjunction with the **show rip stats current** command.

Mode

```
root
```

Syntax

```
reset rip stats
```

Example

```
cli:192.168.208.3:root# reset rip stats
cli:192.168.208.3:root#
```

Related Commands

- **show rip stats current** on page 515

rip

Use this command to enter Router Information Protocol (RIP) configuration mode from within router mode.

Mode

```
router
```

Syntax

```
rip
```

Example

```
cli:172.16.19.10:router# root
mode: root
cli:172.16.19.10:root# router
mode: router
cli:172.16.19.10:router# rip
mode: router:rip
cli:172.16.19.10:router:rip#
```

Related Commands

- **router** on page 356

root

Use this command to enter the top-level root mode.

Mode

Any.

Syntax

root

Example

```
cli:172.16.19.10:slot(1/1)# root
mode: root
cli:172.16.19.10:root# show mode
mode: root
cli:172.16.19.10:root#
```

Related Commands

- **show mode** on page 458

route-map

Defines a route map to control and modify RIP and OSPF routing information. A route map is identified by its map-tag.

If you specify a route-map that exists, this command enters configuration mode for that route-map. If you specify a new route-map, then the system creates the route map and enters configuration mode for the new route-map.

You can permit or deny that routes be advertised or learned, or define the properties with which these routes are distributed. You add route-maps to map-lists to create route filters. These route-maps serve as the criteria that determine which routes to match, and specify the action to take for all matching routes.

Mode

To create an import route-map for RIP routes:

```
router:rip:import
```

To create an export route-map for RIP routes:

```
router:rip:export
```

To create an import route-map for OSPF routes:

```
router:ospf:import
```

To create an export route-map for OSPF routes:

```
router:ospf:export
```

Syntax

While in router:ospf:import mode, the syntax is:

route-map <map-tag>

All ospf routes are must be imported and cannot be denied.

For the other supported modes, the syntax is:

route-map <map-tag> {**permit** | **deny**}

Arguments

<map-tag>	Number of this route-map. This number identifies the route-map to the system.
permit	Permits the import of export of matching routes.
deny	Denies, or prevents, the import or export of matching routes.

Example

```
cli:172.16.19.10:root# router rip import
mode: router:rip:import
cli:172.16.19.10:router:rip:import# route-map 1 permit
cli:172.16.19.10:router:rip:import:route-map(1)# show route-map

ID  Description Route Address   Route Mask      Peer Address    Peer Mask
--- ----------- --------------- --------------- --------------- ---------------
  1                     0.0.0.0         0.0.0.0         0.0.0.0         0.0.0.0

row count: 1
cli:172.16.19.10:router:rip:import:route-map(1)#
```

Related Commands

- **map-list** on page 251
- **match** on page 254
- **override** on page 309
- **show map-list** on page 456
- **show route-map** on page 517

router

Use this command to enter router configuration mode.

Mode

Any.

Syntax

```
router
```

Example

```
cli:172.16.19.10:root# router
mode: router
cli:172.16.19.10:router#
```

Related Commands

- **ospf** on page 306
- **rip** on page 352
- **show mode** on page 458

router-id

Configures the OSPF router ID. The OSPF router ID uniquely identifies the router to other routers within an autonomous system.

In broadcast networks, if the priority values of the routers involved in the designated router election are equal, the router ID determines the designated router. If two or more routers have the same priority, the router with the highest router ID is elected as the designated router for the area.

Note the following:

- The router ID must be unique for every router in the autonomous system.
- You cannot specify 0.0.0.0 or 255.255.255.255 as router IDs.

Mode

```
router:ospf
```

Syntax

```
router-id <ip-address>
```

Arguments

<ip-address>	The router ID that you want to assign to the system, defined in IP address format.

Example

```
cli:172.16.19.10:root# router ospf
mode: router:ospf
cli:172.16.19.10:router:ospf# router-id 201.1.1.1
cli:172.16.19.10:router:ospf# show ospf
Admin Status                          Enabled
TOS Support                             False
Router Id                           201.1.1.1
ABR Status                              False
ASBR Status                             False

Report ospf-nbr-state                 Enabled
Report ospf-virt-nbr-state            Enabled

cli:172.16.19.10:router:ospf#
```

Related Commands

- **asbr** on page 54
- **no asbr** on page 263
- **ospf area** on page 307
- **show ospf** on page 489

router ospf

Use this command to enter router ospf configuration mode. Within this mode you can configure global OSPF parameters, and enter import or export modes to create OSPF route filters.

Mode

Any.

Syntax

```
router ospf
```

Example

```
cli:172.16.19.10:root# router ospf
mode: router:ospf
cli:172.16.19.10:router:
```

Related Commands

- **asbr** on page 54
- **export** on page 166
- **import** on page 179
- **no asbr** on page 263
- **ospf area** on page 307
- **report** on page 348
- **router-id** on page 357

router rip

Use this command to enter router RIP mode. Within this mode you can configure RIP import and export filters.

Mode

Any.

Syntax

```
router rip
```

Example

```
cli:172.16.19.10:root# router rip
mode: router:rip
cli:172.16.19.10:router:rip#
```

Related Commands

- **export** on page 166
- **import** on page 179

save

Use this command to persist (save) system configuration for all active system slots. Active slots are slots that have operational system modules installed.

Mode

```
slot(<c/s>)
root
```

Syntax

save

Example

```
cli:172.16.19.10:root# save
Saving slot: 1/1 ..
Save request passed for slot 1/1
Saving slot: 1/3 ...
Save request passed for slot 1/3
Saving slot: 1/8 ...
Save request passed for slot 1/8
Saving slot: 1/11 .....
Save request passed for slot 1/11
Saving slot: 1/13 .
Save request passed for slot 1/13
cli:172.16.19.10:root#
```

Related Commands

- **root** on page 353
- **slot** on page 555

set paging

Use this command to toggle screen paging on or off. When paging is on, displays that extend beyond the screen are stopped after the screen is filled and continuation of the display is indicated with:

```
- more-
```

You can then hit any key to continue drawing the display.

When paging is off, the entire screen is drawn without interruption.

Mode

Any.

Syntax

```
set paging {on | off}
```

Arguments

on	Enables paging of the display. After screen is filled, the user can hit any key to continue.
off	Displays are drawn without interruption, even if the resulting output extends beyond the current window.

Related Commands

- **set prompt** on page 364

set password

Modifies the password for the current account.

Mode

Any.

Syntax

set password <new-password>

Arguments

<new-password>	New password for the current account.

Example

The following example changes the root password to adcbas:

```
cli:172.16.19.10:root# set password adcbas
Password for 'root' was successfully changed
cli:172.16.19.10:root#
```

Related Commands

- **account** on page 621

set prompt

Use this command to change the CLI prompt. By default, the prompt displays both the IP address of the management module and the current mode. You can configure the prompt so this information is not displayed.

Mode

Any.

Syntax

```
set prompt [mode]
```

Arguments

mode	Configures the prompt to display the IP address of the management module and the current mode.

Example

The following example removes the IP address and current mode from the prompt, then sets the prompt back to its default display.

```
cli:172.16.19.10:root# set prompt
cli# set prompt mode
cli:172.16.19.10:root#
```

set time

Sets the system time. If you set the system time, and you are using an external provisioning system, make sure that the time you set is synchronized with that system.

Mode

Any.

Syntax

set time <string>

Arguments

<string>	A text string that specifies system time in the following format: "<day> <month> <date> <time> <standard> <year>" The string must be enclosed in quotation marks. For example: "Wed Sep 05 13:09:01 EDT 2001" Use the **show time** command to display system time. This command displays system time in the same format in which you must specify system time.

Example

```
cli:192.168.208.3:root# show time
Host system time:   Wed Sep 05 13:18:02 EDT 2001
cli:192.168.208.3:root# set time "Wed Sep 05 13:18:20 EDT 2001"
Setting system time to:  Wed Sep 05 13:18:20 EDT 2001
Host system time:   Wed Sep 05 13:18:20 EDT 2001
cli:192.168.208.3:root# show time
Host system time:   Wed Sep 05 13:18:28 EDT 2001
cli:192.168.208.3:root#
```

Related Commands

- **show time** on page 541

set timeout

Sets the timeout for idle CLI sessions, in minutes.

Mode

Any.

Syntax

```
set timeout <minutes>
```

Arguments

<minutes>	Specifies the timeout for idle CLI sessions, in minutes. Values range from 0 to 3679200.

Example

```
cli:192.168.208.3:root# set timeout 15
Setting CLI timeout period to 15 minutes
cli:192.168.208.3:root# show timeout
Time Out: 15 Minutes
```

Related Commands

- **show timeout** on page 542

shared-secret

Sets a shared secret on the current CMTS interface. The shared secret is shared by both the provisioning server and the CMTS. You must configure the same shared secret for both the CMTS and the provisioning server utilized by that CMTS. The CMTS uses the shared secret to verify that the cable modems received their configurations through a trusted server.

Use the no form of the command to remove the shared secret.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
shared-secret [ascii] <string>

no shared-secret
```

Arguments

ascii	Specify this keyword when entering a text-string as a shared secret. If you enter a string without this keyword, you must enter a hex string.
<string>	Shared secret string. If you enter a hex string, the length of the hex string must be even. Use empty quotes (" ") to remove the shared secret.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# shared-secret 123456
cli:192.168.208.3:interface:cable:csi(1/1/1)# show shared-secret
Shared Key
     ASCII: "¦4V"
     HEX:   "12:34:56"
Shared Key Enabled                      Yes
```

Related Commands

- **show shared-secret** on page 521

show aaa

This command displays the host server's current configuration for network access authentication.

Mode

root

Syntax

show aaa

Example

This example shows that TACACS+ is enabled for network access security authentication.

```
cli:192.168.208.3:root# show aaa
aaa authentication login default tacacs+
cli:192.168.208.3:root#
```

Related Commands

- **aaa authentication** on page 44
- **radius-server** on page 346
- **show radius-server** on page 510
- **show tacacs-server** on page 540
- **tacacs-server** on page 585

show access-class

Displays all access-lists applied to the current interface.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
show access-class
```

Example

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# show access-class

Access List      Priority Row
List   Direction          Status
Number
------ --------- -------- ------
     1        in        2      1

row count: 1
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **access-class** on page 45
- **access-list** on page 47
- **no access-class** on page 260

show access-list

Displays all access-lists defined on the system; use the rule number argument to display a specified access list.

The system keeps a central pool of access lists that you create using the **access-list** command. You can then select from this central pool of access lists when applying them to interfaces using the **access-class** command.

Mode

```
root
interface:cable:csi(<c/s/i>)
```

Syntax

```
show access-list <rule-number>
```

Arguments

<rule-number>	Rule number of the access list that you want to display.

Example 1

The following example displays access list 1 on interface 1/1/1:

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# show access-list 1

Ac Rule Prot IP Source       Mask Source     Start End   Estab TOS ChTOS
             IP Dest         Mask Dest       Port  Port  Sync  Msk
-- ---- ---- --------------- --------------- ----- ----- ----- --- ------
DE    1  tcp         0.0.0.0 255.255.255.255     0 65535 False   0      0
                    0.0.0.0 255.255.255.255     0 65535 False   0
.. .... .... ............... ............... ..... ..... ..... ... ......

row count: 1
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Example 2

The following example displays all access lists defined on the system:

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# show access-list

List  Ac Rule Prot IP Source       Mask Source     Start End   Estab TOS ChTOS
                   IP Dest         Mask Dest       Port  Port  Sync  Msk
----- -- ---- ---- --------------- --------------- ----- ----- ----- --- ------
    1 DE    1  tcp         0.0.0.0 255.255.255.255     0 65535 False   0      0
                        0.0.0.0 255.255.255.255     0 65535 False   0
..... .. .... .... ............... ............... ..... ..... ..... ... ......
    2 PE    1   ip         0.0.0.0 255.255.255.255             False   0      0
                        0.0.0.0 255.255.255.255             False   0
..... .. .... .... ............... ............... ..... ..... ..... ... ......

row count: 2
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **access-class** on page 45
- **access-list** on page 47
- **no access-list** on page 262

show admission-control

Displays the current admission control status (enabled or disabled) for the cable interface.

Mode

```
interface:<type>:csi(<c/s/i>):#
```

Syntax

```
show admission-control
```

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show admission-control
CMTS Admission Control:              Disable
```

Related Commands

- **admission-control** on page 51

show alarm-throttle

Displays the following parameters:

- Alarm delivery interval
- Alarm threshold, which is the maximum number of alarms to deliver during an alarm delivery interval

Mode

`root`

Syntax

`show alarm-throttle`

Example

```
cli:192.168.208.3:root# show alarm-throttle
Alarm Delivery Interval : 10
Max Alarms per Interval : 400
```

Related Commands

- **alarm-throttle** on page 52

show arp

Displays the Address Resolution Protocol (ARP) cache for the current interface. Both dynamic and static ARP entries are displayed.

Mode

```
interface <type> c/s/i
```

Syntax

```
show arp
```

Example

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# show arp

IP Address       MAC Address        Type
---------------- ------------------ ----------
   192.168.19.51 00:90:83:36:82:f1     dynamic
   192.168.19.52 00:90:83:36:82:ee     dynamic
   192.168.19.53 00:90:96:00:39:f9     dynamic
   192.168.19.54 00:90:96:00:39:7f     dynamic
   192.168.19.55 00:90:96:00:29:71     dynamic
   192.168.19.56 00:10:95:01:ef:d8     dynamic
   192.168.19.57 00:10:95:01:f0:05     dynamic
   192.168.19.58 00:90:83:32:9f:8c     dynamic

row count: 14
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **add arp** on page 50
- **clear arp-cache** on page 83
- **del arp** on page 138
- **no proxy-arp** on page 292
- **proxy-arp** on page 343

show arp timeout

Displays the timeout for dynamic ARP cache entries associated with an interface. The timeout can be set with the **arp timeout** command.

Mode

```
interface:<type>:csi(<c/s/i>):#
```

Syntax

```
show arp timeout
```

Example

```
cli:192.168.208.3:interface:cable:csi(1/2/1)# show arp timeout
ARP Aging                              Enabled
ARP Timeout                                600

cli:192.168.208.3:interface:cable:csi(1/2/1)# arp timeout 700
cli:192.168.208.3:interface:cable:csi(1/2/1)# show arp timeout
ARP Aging                              Enabled
ARP Timeout                                700
```

Related Commands

- **arp timeout** on page 53

show aux-device

Displays the current power supply and fan tray fault report configuration, including which alarms are sent out over the alarms out DB-15 connector on the rear chassis panel. The command also displays clock sources for backplane clock A and backplane clock B.

Mode

root

Syntax

show aux-device {**ac-monitor** | **backplane-clocks** | **db15** | **dc-monitor** | **fan-rotation** | **fan-temp** | **faults** | **ps-temp**}

Arguments

ac-monitor	Displays the assertion level configured for monitoring of AC power.
backplane-clocks	Displays the sources for backplane clock A and backplane clock B. Clock sources are none, bits-a, bits-b, and internal. The command output also indicates whether the Stratum-3 oscillator is installed.
db15	Shows the alarms configured to be sent out over the alarms out DB-15 connector on the rear chassis panel. Note that a value of "enabled" indicates that the alarm is configured to be sent out over the DB-15 connector; a value of "disabled" indicates that it will not be sent over the DB-15 connector.
dc-monitor	Displays the assertion level configured for monitoring of DC power.
fan-rotation	Displays the assertion level configured for monitoring the rotation of fans within the fan tray unit.
fan-temp	Displays the assertion level configured for the monitoring of the fan tray temperature.
faults	Displays the assertion level currently configured for all devices.
ps-temp	Displays the assertion level configured for the monitoring of power supply temperature.

Example

```
cli:172.16.19.10:root# show aux-device faults
Assert Ps Temp Fault             active-high
Assert Ps AC Fault               active-high
Assert Ps DC Fault               active-high
Assert Fan Temp Fault            active-high
Assert Fan Rotation Fault        active-high
```

Related Commands

- **aux-device ac-monitor** on page 55
- **aux-device backplane-clock-a** on page 56
- **aux-device backplane-clock-b** on page 57
- **aux-device db15 alarm** on page 58
- **aux-device dc-monitor** on page 62
- **aux-device fan-rotation** on page 63
- **aux-device fan-temp** on page 64
- **aux-device ps-temp** on page 65

show boot

Displays the current boot configuration — enabled or disabled.

When you issue the command in root mode, you must specify the slot keyword and the <c/s> slot identifier. When you issue the command in slot mode, you do not have to specify any arguments.

Mode

```
root
slot <c/s>
```

Syntax

```
show boot [slot <c/s>]
```

Arguments

slot <c/s>	Specifies the slot for which you want to obtain boot status.

Example

```
cli:172.16.19.10:root# slot 1/1
mode: slot(1/1)
cli:172.16.19.10:slot(1/1)# show boot
Slot State                  enabled

cli:172.16.19.10:slot(1/1)#
```

Related Commands

- **boot** on page 67
- **slot** on page 555

show bootp-policy

Displays BOOTP policies for an interface. Issuing this command without any arguments displays all BOOTP policies configured for an interface.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

show bootp-policy [{<index> | **default**}]

Arguments

<index>	Specifies the index number of a BOOTP policy, which displays details on that policy only.
default	Displays details on the default BOOTP policy.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show bootp-policy

row count: 2

Index Description    Mac Address       Mask              Action Server List
----- -------------- ----------------- ----------------- ------ -------------
    2                00:90:11:00:00:00 ff:ff:ff:00:00:00 permit 202.199.1.1
                                                                  202.199.1.2
                                                                  202.199.1.3
                                                                  202.199.1.4
    3                4a:01:11:00:00:00 ff:ff:ff:00:00:00   deny

Default bootp policy:
Policy Action                           deny
Policy Server List

cli:192.168.208.3:interface:cable:csi(1/1/1)# show bootp-policy 2
Index                                     2
Description
Mac Address                00:90:11:00:00:00
Mask                       ff:ff:ff:00:00:00
Action                                 permit
Server List                202.199.1.1 202.199.1.2 202.199.1.3 202.199.1.4

cli:192.168.208.3:interface:cable:csi(1/1/1)#
```

Related Commands

- **bootp-policy** on page 68

show bridge-group

Displays the network-layer bridge groups currently defined on the system. Use this command with no arguments to display all bridge groups defined on the system. Pass the command a bridge group number or name to show a specified bridge group.

Mode

Any.

Syntax

show bridge-group [{<group-number> | <group-name>}]

Arguments

<group-number>	For bridge groups identified by number, specify the number of the bridge group that you want to display.
<group-name>	For bridge groups identified by text string, specify the name of the bridge group that you want to display.

Example 1

The following example displays group number 1:

```
cli:172.16.19.10:root# show bridge-group 1

Bridge Group: 1
C/S/I:        129/1/0

Chassis    Slot       Interface
---------- ---------- ----------
         1          1          1
         1         11          1

row count: 2
```

Example 2

The following example displays the bridge group named LAN_1:

```
cli:172.16.19.10:root# show bridge-group LAN_1
Bridge Group: LAN_1
```

Example 3

The following example displays all bridge groups defined on the system:

```
cli:172.16.19.10:root# show bridge-group

Bridge Group: 1

Chassis    Slot       Interface
---------- ---------- ----------
         1          1          1
         1         11          1

row count: 2
Bridge Group: LAN_1

cli:172.16.19.10:root#
```

Related Commands

- **bridge-group** on page 70

show bridge-timeout

Displays aging and reply timers for bridge group broadcast flows.

Mode

To displays timers for all bridge groups on all interfaces:

`root`

To display timers for a specific bridge group:

`interface:bridge-group(<group>)`

To display timers for a specific interface:

`interface:cable:csi(<c/s/i>)`

Syntax

`show bridge-timeout`

Example

```
cli:interface:cable:csi(1/1/1)# show bridge-timeout
Reply Timeout                              9
Aging Timeout                             15
```

Related Commands

- **bridge-timeout** on page 74

show buffers

Displays buffer usage for the current module. Note that this command does not show buffer usage for the control module.

Mode

```
slot(c/s)
```

Syntax

```
show buffers
```

Example

```
cli:192.168.208.3:slot(1/1)# show buffers
index[0] = 1
index[1] = 1
index[2] = 0
index[3] = 2
Buffer Pool Size                        1024
Buffer Allocated                         151
Buffer Available                         873
```

The fields are described as follows:

Field	Description
index[0]	The chassis ID.
index[1]	The slot ID.
index[2]	The interface ID of the module.
index[3]	The logical port ID of the module.
Buffer Pool Size	The total number of memory buffers for the module. The total equals the sum of the allocated buffers and available buffers.
Buffer Allocated	The total number of memory allocated memory buffers.
Buffer Available	The total number of available (free) memory buffers.

show chassis

Displays details on the local chassis or a remote chassis.

Mode

`root`

Syntax

`show chassis` {**`local`** | <ip-address>}

Arguments

local	Displays details on the local chassis.
<ip-address>	Displays details on the chassis with the specified IP address.

Example

```
cli:root# show chassis local

Multi Chassis Service : disable
Host Name   : Tech1
IP Address  : 192.168.208.3
Group Name  : group1
Version     : 3.0.6 CPM3.0 21 2001_07_17_1036
Description : null
```

Related Commands

- **chassis** on page 77

show chassis-config

Shows the current chassis configuration, including:

- **chassis-number** — Number assigned to the chassis (default chassis number is 101).
- **ChassisID —** Identifies the chassis.
- **clusterID** — Displays the chassis cluster ID. The default is 0, which is the recommended value.
- **Manager** — Indicates the slot number of the primary management module and the slot number of the secondary management module. The system supports two management modules for Management Module Redundancy.
- **Scope** — One of the following:

  Chassis — The management module can be used to configure only the current system.

  Cluster — The scope of this management module is cluster-wide. This is the default configuration as the current release supports a single-chassis configuration only.

Mode

Any.

Syntax

```
show chassis-config
```

Example

```
cli:172.16.19.10:root# show chassis-config
Chassis Number:         101
Chassis Id:             1
Cluster Id:             0
Primary Manager Slot:   13
Secondary Manager Slot: None
Scope:                  Cluster
```

Related Commands

- **chassis-config** on page 78

show chassis-fault status

Displays which Cuda 12000 chassis alarms are enabled and disabled. Enabled alarms are listed as "okay." Disabled alarms are listed as "disabled." If a fault has caused an alarm to generate, the status is "faulted."

Mode

`root`

Syntax

`show chassis-fault status`

Example

```
cli:192.168.244.212:root# chassis-fault backplane-power local-pwr-a
cli:192.168.244.212:root# show chassis-fault status

Chassis Fault Status

Bits A Fault                        disabled
Bits B Fault                        disabled
Backplane System Fault              disabled
Backplane Temp Fault                disabled
Backplane Power Fault                   okay
Backplane Power A Fault             disabled
Backplane Power B Fault             disabled
Red Alarm Fault                     disabled
Blue Alarm Fault                    disabled
Yellow Alarm Fault                  disabled
Processor Temp Fault                disabled
Ps Temp Fault                       disabled
Ps AC Fault                         disabled
Ps DC Fault                         disabled
Fan Temp Fault                      disabled
Fan Rotation Fault                  disabled
Local Pwr A Fault                       okay
Local Pwr B Fault                   disabled
```

Related Commands

- **chassis-fault** on page 80

show clock-synchronization

Displays the status of clock synchronization on a module.

Mode

slot

Syntax

show clock-synchronization

Example

```
cli:192.168.244.212:slot(1/4)# show clock-synchronization
Clock Synch Status                unSupported
Card Drive Clock_A          xxxxxxxxxxxxxxx
Card Drive Clock_B          xxxxxxxxxxxxxxx
```

Related Commands

- **aux-device backplane-clock-a** on page 56
- **aux-device backplane-clock-b** on page 57

show cm-filter

Displays filters for cable modem and CPE traffic. Issuing the command with no arguments displays all filters on the CMTS.

Mode

`root`

Syntax

`show cm-filter` [<group-id> [<index>]]

Arguments

<group-id>	Specifies the group ID (a number between 1 and 60) that identifies the filter group. This displays all filters in the group.
<index>	Specifies the index number of a filter within the group. This displays details on the specified filter only.

Example

```
cli:192.168.208.3:root# show cm-filter 2 1
Group                                    2
Index                                    1
Src Address               0.0.0.0
Src Mask                  0.0.0.0
Dest Address              144.133.1.0
Dest Mask                 255.255.255.0
Protocol                              tcp
TOS                       00
TOS Mask                  00
Action                             deny
Matches                               0
Source Port                       65536
Destination Port                  65536
TCP Flag Values
TCP Flag Mask
```

Related Commands

- **cm-filter** on page 88

show cm-filter-default

Displays the default cable modem and CPE filter groups.

Mode

Any.

Syntax

show cm-filter-default

Example

```
cli:root# show cm-filter-default
CPE DS Filter Group                    0
CPE US Filter Group                    1
CM DS Filter Group                     0
CM US Filter Group                     0
```

Related Commands

- **cm-filter-default cm downstream** on page 92
- **cm-filter-default cm upstream** on page 94
- **cm-filter-default cpe downstream** on page 96
- **cm-filter-default cpe upstream** on page 98

show cm-offline

Displays the parameters that the CMTS uses to track and maintain statistics for offline cable modems on the current interface:

- Cable Modem Offline Timer — Displays the number of days that the CMTS tracks offline cable modems. You set the value of this parameter using the **cm-offline timer** command.
- Cable Modem Stats Persist — Displays whether the ability of the CMTS to maintain statistics for cable modems that go offline is enabled or disabled. You set the value of this parameter using the **cm-offline persist** command.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
show cm-offline
```

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show cm-offline
Cable Modem Offline Timer                 35
Cable Modem Stats Persist              enabled
```

Related Commands

- **cm-offline persist** on page 102
- **cm-offline timer** on page 103

show controllers

Displays SONET defect and alarm information for a specified Packet Over SONET (POS) interface.

Mode

`root`

Syntax

`show controllers pos` <c/s/i>

Arguments

<c/s/i>	Interface for which you want to display alarm and defect information identified in terms of chassis/slot/port.

Example

```
cli# root
mode: root
cli# show controllers pos 1/3/1
--from itable ----------------------------------
Interface Type                              39
POS 1/3/1 (line protocol)                closed

--from bassonetline table-----------------------
Section: LOF                                 0
Section: LOS                                 0
Section: BIP(B1)                             0
Line: AIS                                    0
Line: RDI                                    0
Line: FEBE(M1)                               0
Line: BIP(B2)                                0
Clock Recovery: LOC                          0

--from bassonetpath table-----------------------
Path: AIS                                    0
Path: RDI                                    0
Path: FEBE(G1)                               0
Path: BIP(B3)                                0
Path: LOP
--from bassonetmediumconfig table-----------------

--from bassonetmediumstatus table----------------
Framing                                   Sonet

--from bassonetalarmreport-table------------------
Report alarms for B1                     enabled
Report alarms for B2                     enabled
Report alarms for B3                     enabled
Report alarms for LAIS                   enabled
Report alarms for LRDI                  disabled
Report alarms for PAIS                  disabled
Report alarms for PLOP                   enabled
Report alarms for PRDI                  disabled
Report alarms for SD-BER                disabled
Report alarms for SF-BER                 enabled
Report alarms for SLOF                   enabled
Report alarms for SLOS                   enabled
Threshold for SD report                        7
Threshold for SF report                        4

--from bassonetalarm-table-----------------------
```

```
Local alarm active now                  None
Remote alarm active now                 None
B1 errors occurring now                false
B2 errors occurring now                false
B3 errors occurring now                false
LAIS errors occurring now              false
LRDI errors occurring now              false
PAIS errors occurring now              false
PLOP errors occurring now              false
PRDI errors occurring now              false
SD-BER errs occurring now              false
SF-BER errs occurring now              false
SLOF errors occurring now              false
SLOS errors occurring now              false
```

Related Commands

- **interface** on page 181
- **no ppp username** on page 291

show cpe-control

Displays subscriber management default settings for cable modems.

Mode

Any.

Syntax

show cpe-control

Example

```
cli:interface:cable:csi(1/1/1)# show cpe-control
MAX IP                                    16
Active                                 False
Learnable                               True
```

Related Commands

- **cpe-control active** on page 130
- **cpe-control learnable** on page 131
- **cpe-control max-ip** on page 132

show cpu-utilization

Displays CPU utilization for both the network processor and Pentium processor on installed modules. All forwarding modules utilize a network processor; only DOCSIS/EuroDOCSIS modules and route server modules utilize the additional Pentium processor.

After you enable CPU utilization using this command, you can then use the **show cpu-utilization** feature to display CPU usage for these module processors.



Use the* no cpu-utilization *command to turn this feature off.

Mode

```
slot(<c/s>)
```

Syntax

```
show cpu-utilization {1 | 2} {h | d | m}
```

Arguments

1	Displays CPU utilization for the network processor. Applicable to all forwarding modules.
2	Displays CPU utilization for the on-board Pentium processor. Applicable only to DOCSIS/EuroDOCSIS modules, and modules configured as the route server.
h	Displays CPU usage for a 1 hour duration in 1 minute intervals.
d	Displays CPU usage for a 1 day duration in 1 hour intervals.
m	Displays CPU usage for a 1 minute duration in 5 second intervals.

Example

```
cli:172.16.19.10:root# slot 1/1
mode: slot(1/1)
cli:172.16.19.10:slot(1/1)# show cpu-utilization 1 m

Cpu Utilization, Duration: 1 Minute in 5 second intervals

row count: 12

Time       Avg CPU
(Seconds)  Usage
---------- ----------
         5          5
        10          3
        15          2
        20          2
        25          2
        30          2
        35          2
        40         15
        45          4
        50          4
        55          2
        60          5

cli:172.16.19.10:slot(1/1)# show cpu-utilization 2 h

Cpu Utilization, Duration: 1 Hour in 1 minute intervals

row count: 60
row count: 60

Time       Avg CPU    Max CPU    Max CPU
(Minutes)  Usage      Usage      Usage Time
---------- ---------- ---------- ----------
         1         13         14          2
         2         13         15          3
         3         13         14          6
         4         13         14          4
         5         13         14         11
         6         12         14          6
         7         13         14          7
         8         14         18          7
         9         13         14          8
        10         13         14          8
        11         13         14          8
        12         13         14         12
        13         13         14         10
        14         13         14         11
        15         12         13         12
 --More--
```

show db-connect

Displays provisioning server access information.

Mode

Any.

Syntax

show db-connect

Example

```
cli:172.16.19.10:root# show db-connect
ShowDBConnect: showing dbconnect info
ProvServerId =1
ProvInfoLdapServerIpAddr =127.0.0.1
ProvInfoLdapServerPort =389
ProvInfoLdapServerUserName =cn=Directory Manager
ProvInfoLdapServerPassword =********
cli:172.16.19.10:root#
```

Related Commands

- **db-connect** on page 136
- **no db-connect** on page 267

show dhcp-authority

Displays DHCP authority ranges for the interface. Issuing the command with no arguments displays all ranges.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

show dhcp-authority [{<index> | **default**}]

Arguments

<index>	Specifies the index number of a DHCP authority range, which displays details on that range only.
default	Displays details on the default DHCP authority range.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show dhcp-authority


Range Number     Lower Range      Upper Range      Status
---------------- ---------------- ---------------- ------

DHCP Authority Status                disable

cli:192.168.208.3:interface:cable:csi(1/1/1)# dhcp-authority 4 start
201.1.2.140 end 201.1.2.160
cli:192.168.208.3:interface:cable:csi(1/1/1)# show dhcp-authority

row count: 1

Range Number     Lower Range      Upper Range      Status
---------------- ---------------- ---------------- ------
               4 201.1.2.140      201.1.2.160           1

DHCP Authority Status                disable

cli:192.168.208.3:interface:cable:csi(1/1/1)# dhcp-authority enable
cli:192.168.208.3:interface:cable:csi(1/1/1)# show dhcp-authority

row count: 1

Range Number     Lower Range      Upper Range      Status
---------------- ---------------- ---------------- ------
               4 201.1.2.140      201.1.2.160           1

DHCP Authority Status                 enable
```

Related Commands

- **dhcp-authority** on page 139

show dhcp-policy

Displays DHCP policies configured for the current interface. Issuing the command with no arguments displays all policies.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
show dhcp-policy [{<index> | default}]
```

Arguments

<index>	Specifies the index number of a DHCP policy, which displays details on that policy only.
default	Displays details on the default DHCP policy.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show dhcp-policy
Default dhcp policy:
Policy Action                        permit
Policy Server List
Forward Internal                     enable
```

Related Commands

- **dhcp-policy** on page 142

show dhcp-relay

Displays DHCP relay information for the current cable interface.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
show dhcp-relay [counters]
```

Arguments

counters	Use this argument to include DHCP Relay statistics in the display.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# show dhcp-relay

dhcp-relay                              enable
Add Agent Options                       enable
Drop Mismatch                          disable
Max. Pkt. Len.                             576
Relay Mode                             replace



Server Address
----------------

giAddresses:
CM                                  201.0.0.1
CPE                                 202.0.0.1
MTA                                   0.0.0.0
```

Related Commands

- **dhcp-relay** on page 145
- **no dhcp-relay** on page 268

show downstream

Displays the downstream configuration parameters for the current CMTS interface.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

show downstream

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show downstream
C/S/P                            1 / 1 / 2 / 2
Frequency                                507.0  (MHz)
Interleave                  taps32Increment4
Modulation                               qam64
ChannelWidth                                 6  (MHz)
ChannelPower                              50.0  (dBmV)
AnnexType                               Annex B

Symbol Rate                              5056941  (baud)

Admin Status                                  up
Operational Status                            up

Out Octets                           3666257239
Out Unicast Packets                     3407902
Out Multicast Packets                 464122469
Out Broadcast Packets                     19875
Out Errors                                    0
Out Discards                                  0
```

Related Commands

- **downstream frequency** on page 148
- **downstream interleave-depth** on page 150
- **downstream modulation** on page 151
- **downstream transmit-power** on page 155

show event-config

Displays the DOCSIS 1.1 event transmission, reporting, and syslog parameters. If you do not specify any arguments, the command displays all of the parameters.

Mode

`root`

Syntax

`show event-config {throttle | reporting | syslog}`

Arguments

throttle	Displays the parameters that control the pace of event transmission (threshold, interval, admin, throttle inhibited). In the command output, note that the Throttle Inhibited field displays True if one of the following conditions is met: ■ Event Administrative Status is set to inhibited. ■ Event Administrative Status is set to stopAtThreshold and the threshold has been reached. Otherwise, this field displays False.
reporting	Displays the event classes and the reporting action for each class.
syslog	Displays the IP address of the syslog server.

Example

```
cli:192.168.208.3:root# show event-config
Event Throttle Parameters
-------------------------
Threshold                                    0
Interval                                     1
Admin Status                    unconstrained
Throttle Inhibited                      False
Event Reporting Priorities
--------------------------
row count: 8
Priority    Action
----------- ------------------
  emergency local|traps|syslog
      alert local
   critical local|traps|syslog
      error local|traps|syslog
    warning local|traps|syslog
     notice local|traps|syslog
information none
      debug none
Syslog Server                     133.132.1.1
cli:192.168.208.3:root# show event-config throttle
Event Throttle Parameters
-------------------------
Threshold                                    0
Interval                                     1
Admin Status                    unconstrained
Throttle Inhibited                      False

cli:192.168.208.3:root# show event-config reporting
Event Reporting Priorities
--------------------------
row count: 8
Priority    Action
----------- ------------------
  emergency local|traps|syslog
      alert local
   critical local|traps|syslog
      error local|traps|syslog
    warning local|traps|syslog
     notice local|traps|syslog
information none
      debug none
cli:192.168.208.3:root# show event-config syslog
Syslog Server                     133.132.1.1
```

Related Commands

- **event-config reporting** on page 159
- **event-config syslog** on page 162
- **event-config throttle** on page 163

show event-log

Displays the contents of the event log.

Mode

root

Syntax

show event-log

Example

```
cli:192.168.220.206:root# show event-log
row count: 133
Index  First Time Last Time  Counts     Level      ID         Text
------ ---------- ---------- ---------- ---------- ---------- ----------
     1 2000-12-31 2000-12-31          1   critical 2147483652 CMTS/CM
       ,21:1:40.0 ,21:1:40.0                                  Down -
       ,455:0     ,455:0                                      ifIndex =
                                                              8781825
     2 2000-12-31 2000-12-31          1   critical 2147483649 Card Down
       ,21:31:40. ,21:31:40.                                  - 1/1/1
       0,455:0    0,455:0
     3 2001-1-1,1 2001-3-6,1       1264   critical 2147483652 CMTS/CM
       :20:0.0,45 :26:40.0,4                                  Down -
       5:0        55:0                                        ifIndex =
                                                              8781825
     4 2000-12-31 2000-12-31          2   critical 2147483652 CMTS/CM
       ,19:28:20. ,19:28:20.                                  Down -
       0,455:0    0,455:0                                     ifIndex =
                                                              8781825
     5 2000-12-31 2000-12-31          5   critical 2147483652 CMTS/CM
       ,19:36:40. ,23:46:40.                                  Down -
       0,455:0    0,455:0                                     ifIndex =
--More--
```

The **show event-log** command output displays these fields of information about each event:

Field	Description
Index	The number of the event in the log. This number is used to order the events in the log.
First Time	The time that the log entry was created.
Last Time	The time that the last event associated with the log entry occurred. In some cases, multiple events can be associated with a single log entry. This tends to happen when duplicate events are reported. However, when only one event is reported, then one event is associated with an entry, which means that the First Time and Last Time values are the same.
Counts	The number of consecutive event instances that this event entry reports. The count starts at 1 when the entry is created and increments by one for each subsequent duplicate event.
Level	The event's class (emergency, alert, critical, error, warning, notice, info, debug).
ID	An internal event identifier. The Text field describes the event associated with this identifier.
Text	Brief description of the event.

Related Commands

- **event-log clear** on page 165

show filter-aging

Displays IP packet filter aging parameters for all interfaces in the current slot.

Mode

`slot`

Syntax

`show filter-aging`

Example

```
cli:192.168.208.3:slot(1/1)# show filter-aging
IP Filter Aging In                    enable
IP Filter Aging In Rate                 4096
IP Filter Aging Out                   enable
IP Filter Aging Out Rate                4096
```

Related Commands

- **filter-aging** on page 167

show flap-list

Displays flap list parameters and other information for the current CMTS interface.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
show flap-list [{control | ext | sortbyflapcnt | sortbytime}]
```

Arguments

{**control** | **ext** | **sortbyflapcnt** | **sortbytime**}	Specifies one of the following values: ■ control – Displays information on cable modems that are currently in the flap list table. ■ ext – Displays statistics that are currently in the flap list table. ■ sortbyflapcnt – Sorts the flap list display output by flap count. ■ sortbytime – Sorts the flap list display output by time.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show flap-list control
Max Table Size                          8191  (rows)
Aging Threshold                           60  (days)
Insert Time Threshold                 604800  (secs)
Power Adjustment Threshold                 3  (dBmV)
```

Related Commands

- **flap-list aging** on page 169
- **flap-list clear** on page 170
- **flap-list insertion-time** on page 172
- **flap-list power-adj-threshold** on page 174
- **flap-list size** on page 175

show flash

Displays the configuration and boot settings for the module in the current slot.

Mode

`slot(c/s)`

Syntax

`show flash {config | image}`

Arguments

config	Displays the persisted configuration for the module in the current slot.
image	Displays the boot settings for module in the current slot.

Example

```
cli:172.16.19.10:root# slot 1/11
mode: slot(1/11)
cli:172.16.19.10:slot(1/11)# show flash config

Entries on processor 2:

Index Version Date and Time    File       Transfer Remove   Remove
                               Transfer   status   State
----- ------- ---------------- ---------- -------- -------- ------
    1       0 1980-1-1,0:0:0.0     normal     none   locked normal
    2       0 1980-1-1,0:0:0.0     normal     none   locked normal

row count: 2
cli:172.16.19.10:slot(1/11)#
```

Related Commands

- **boot** on page 67
- **save** on page 361

show host

Displays the IP address of the management module that you are logged into; the IP address assigned to the Ethernet craft port.

Mode

Any.

Syntax

show host

Example

```
cli:172.16.19.10:root# show host
host: 172.16.19.10
cli:172.16.19.10:root#
```

show in-counters

Displays inbound counter statistics for all interfaces installed on the system.

Mode

Any.

Syntax

show in-counters

Example

```
cli:172.16.19.10:root# show in-counters

Interface     Type           In octets  In unicast In multicast In broadcast
------------- -------------- ---------- ---------- ------------ ------------
    1 / 1 / 1   docsCableMac     841298      22174            0            1
    1 / 1 / 3 docsCableUS(1)     502959      13084            0            1
    1 / 1 / 4 docsCableUS(2)     338339       9090            0            0
    1 / 1 / 5 docsCableUS(3)          0          0            0            0
    1 / 1 / 6 docsCableUS(4)          0          0            0            0
    1 / 3 / 1           sonet         0          0            0            0
    1 / 3 / 2      sonetPath          0          0            0            0
    1 / 3 / 3             ppp         0          0            0            0
    1 / 8 / 1       ethernet          0          0            0            0
   1 / 11 / 1       ethernet          0          0            0            0
   1 / 11 / 2       ethernet          0          0            0            0
   1 / 11 / 3       ethernet          0          0            0            0
   1 / 11 / 4       ethernet          0          0            0            0
   1 / 11 / 5       ethernet          0          0            0            0
   1 / 11 / 6       ethernet          0          0            0            0
   1 / 11 / 7       ethernet          0          0            0            0
   1 / 11 / 8       ethernet          0          0            0            0
  129 / 1 / 0     net bridge          0          0            0            0
  129 / 1 / 1     net bridge          0          0            0            0

row count: 19
cli:172.16.19.10:root#
```

Related Commands

- **show out-counters** on page 491

show interface

Displays configuration parameters and statistics for Ethernet and POS interfaces.

Mode

```
interface:ethernet:csi(<c/s/i>)
interface:pos:csi(<c/s/i>)
```

Syntax

```
show interface
```

Example

```
cli:192.168.208.3:interface:ethernet:csi(1/11/1)# show interface
Interface                  1/11/1 Ethernet (100 Mb)
Configured duplex                        auto
Configured speed                         auto
Auto-negotiation                       enable

In octets                                   0
In unicast                                  0
In multicast                                0
In broadcast                                0

Out octets                                  0
Out unicast                                 0
Out multicast                               0
Out broadcast                               0
```

Related Commands

- **duplex** on page 156
- **negotiation auto** on page 259
- **show interface ethernet** on page 435
- **show interface pos** on page 437
- **speed** on page 582

show interface cable <c/s/i> cm-filter-default

Displays the default upstream and downstream filter groups.

Mode

```
root
interface:cable:csi(<c/s/i>)
```

Syntax

show interface cable <c/s/i> **cm-filter-default**

Example

```
cli:root# show interface cable 1/1/1 cm-filter-default
CPE DS Filter Group                          1
CPE US Filter Group                          1
CM DS Filter Group                           1
CM US Filter Group                           1
```

Related Commands

- **cm-filter-default cm downstream** on page 92
- **cm-filter-default cm upstream** on page 94
- **cm-filter-default cpe downstream** on page 96
- **cm-filter-default cpe upstream** on page 98

show interface cable <c/s/i> cpe-control

Displays subscriber management default settings for cable modems.

Mode

Any.

Syntax

show interface cable <c/s/i> **cpe-control**

Example

```
cli:root# show interface cable 1/1/1 cpe-control
MAX IP                                    16
Active                                 False
Learnable                               True
```

Related Commands

- **cpe-control active** on page 130
- **cpe-control learnable** on page 131
- **cpe-control max-ip** on page 132

show interface cable <c/s/i> downstream

Displays downstream port configuration and statistics.

Mode

Any.

Syntax

show interface cable <c/s/i> **downstream**

Example

```
cli:172.16.19.10:root# show interface cable 1/1/1 downstream
C/S/P                          1 / 1 / 2 / 2
Frequency                              507.0  (MHz)
Interleave                 taps32Increment4
Modulation                           qam256
ChannelWidth                              6  (MHz)
ChannelPower                            550  (1/10 dBmV)
AnnexType                           Annex B

Symbol Rate                          5360537  (baud)

Admin Status                              up
Operational Status                        up

Out Octets                         683451855
Out Unicast Packets                    14972
Out Multicacast Packets             15357846
Out Broadcast Packets                    358
Out Errors                                 0
Out Discards                               0

cli:172.16.19.10:root#
```

show interface cable <c/s/i> dynamic-service-stats

Displays the service flow statistics created through a Dynamic Service, initiated by the cable modem or CMTS.

Mode

```
root

interface:cable:csi(<c/s/i>)
```

Syntax

show interface cable <c/s/i> **dynamic-service-stats**

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# root
mode: root
cli:192.168.208.3:root# show interface cable 1/1/1 dynamic-service-stats
Direction                           Outbound
DSA Requests                               0
DSA Responses                              0
DSA Acks                                   0
DSC Requests                               0
DSC Responses                              0
DSC Acks                                   0
DSD Requests                               0
DSD Responses                              0
Dynamic Adds                               0
Dynamic Add Fails                          0
Dynamic Changes                            0
Dynamic Change Fails                       0
Dynamic Deletes                            0
Dynamic Delete Fails                       0
DCC Requests                               0
DCC Responses                              0
DCC Acks                                   0
DCCs                                       0
DCC Fails                                  0

Direction                            Inbound
DSA Requests                               0
DSA Responses                              0
DSA Acks                                   0
DSC Requests                               0
DSC Responses                              0
DSC Acks                                   0
DSD Requests                               0
DSD Responses                              0
Dynamic Adds                               0
Dynamic Add Fails                          0
Dynamic Changes                            0
Dynamic Change Fails                       0
Dynamic Deletes                            0
Dynamic Delete Fails                       0
DCC Requests                               0
DCC Responses                              0
DCC Acks                                   0
DCCs                                       0
DCC Fails                                  0
```

show interface cable <c/s/i> flap-list

Displays the flap list for the specified cable interface.

Mode

Any.

Syntax

show interface cable <c/s/i> **flap-list** [**sortbyflapcnt** | **sortbytime**]

Example

```
cli:172.16.19.10:root# show interface cable 1/1/1 flap-list

Mac Address       Flap  Last Known    Insert Time    Remove Time    Hit   Miss
                  Count State                                     Count Count
----------------- ----- ------------- -------------- -------------- ----- -----
0:10:95:1:ef:d8       4 DhcpDiscRcvd  99-08-03 13:23 99-08-03 13:22  1036    48
0:10:95:1:f0:5        4 DhcpDiscRcvd  99-08-03 13:23 99-08-03 13:22  1036    48
0:10:95:4:a:b7        5 DhcpDiscRcvd  99-08-03 13:23 99-08-03 13:22  1044    64
0:10:95:4:a:bd        5 DhcpDiscRcvd  99-08-03 13:23 99-08-03 13:22  1045    64
0:10:95:4:a:c3        5 DhcpDiscRcvd  99-08-03 13:23 99-08-03 13:23  1044    64
0:10:95:4:a:c4        5 DhcpDiscRcvd  99-08-03 13:23 99-08-03 13:22  1044    64
0:90:83:32:9f:8c      3 DhcpDiscRcvd  99-08-03 13:23 99-08-03 13:20  1027    32
0:90:83:36:82:ee      3 DhcpDiscRcvd  99-08-03 13:22 99-08-03 13:20  1029    32
0:90:83:36:82:f1      3 DhcpDiscRcvd  99-08-03 13:22 99-08-03 13:20  1029    32
0:90:96:0:29:6d      49 DhcpReqRcvd   99-08-03 13:35 99-08-03 13:35  1119   273
0:90:96:0:29:71      21 DhcpDiscRcvd  99-08-03 13:22 99-08-03 13:22  1068   122
0:90:96:0:39:7f      23 DhcpDiscRcvd  99-08-03 13:22 99-08-03 13:22  1079   164
0:90:96:0:39:f9      26 DhcpDiscRcvd  99-08-03 13:22 99-08-03 13:22  1082   132
0:a0:73:69:39:65      3 DhcpDiscRcvd  99-08-03 13:23 99-08-03 13:21  1028    32

row count: 14
cli:172.16.19.10:root#
```

show interface cable <c/s/i> mac

Displays MAC configuration and statistics for the specified cable interface.

Mode

Any.

Syntax

show interface cable <c/s/i> **mac**

Example

```
cli:172.16.19.10:root# show interface cable 1/1/1 mac
Insertion Interval                          10  (centisec)
Invited Ranging Attempts                    16
Sync Interval                                5  (millisec)
UCD Interval                              2000  (millisec)

Hardware Map Timer                        2000  (microsec)
Periodic Ranging Timer                      15  (secs)
Plant Delay                               1600

PLL State                                normal
PLL Value                                     0

Stats:
Admin Status                                 up
Operational Status                           up
In Octets                                628646
In Unicast Packets                        16534
In Multicacast Packets                        0
In Broadcast Packets                          1
In Errors                                    41
In Discards                                   0
Out Octets                           690478794
Out Unicast Packets                       15117
Out Multicacast Packets                15515781
Out Broadcast Packets                       359
Out Errors                                    0
Out Discards                                  0

Invalid Range Requests                        0
Ranging Aborts                                4
Invalid Registration Reque                    0
Failed Registration Reques                    0
Invalid Data Requests                         0
T5 Timeouts                                   0

cli:172.16.19.10:root#
```

show interface cable <c/s/i> modem

Displays cable modem status summary for the selected cable interface.

Mode

Any.

Syntax

show interface cable <c/s/i> **modem** [<mac-address>]



Descriptions of arguments are provided on the reference pages for* show modem** on page 459 ***and* show modem <mac-address>** on page 467.***

Example

```
cli:192.168.208.3:root# show interface cable 1/1/1 modem

row count: 12

MAC Address       IP Address      SID  CID  CPE D:U Power  Timing Modem
                                                   (dbMV)        State
----------------- --------------- ---- ---- --- --- ------ ------ ----------
00:10:95:04:0a:b7 201.1.1.109       1    1   0 1:2      0   2727 Registered
00:90:96:00:39:f9 201.1.1.101       2    1   0 1:2      0   2215 Registered
00:10:95:04:0a:c3 0.0.0.0           3  152   0 1:2     -9   2727 Ranging
00:90:83:36:82:ee 201.1.1.108       4    0   0 1:2      0   1234 RegBpiTek
00:10:95:01:f0:05 201.1.1.103       5    1   0 1:2      0   2210 Registered
00:10:95:04:0a:c4 201.1.1.110       6    1   0 1:2      0   2728 Registered
00:90:96:00:29:6d 201.1.1.104       7    1   0 1:2      0   2218 Registered
00:90:96:00:29:71 201.1.1.112       8    1   0 1:2      0   2216 Registered
00:90:83:32:9f:8c 201.1.1.106       9    1   0 1:2      0   1637 Registered
00:10:95:01:ef:d8 201.1.1.100      10    1   0 1:2      0  10204 Registered
00:90:96:00:39:7f 201.1.1.102      11    1   0 1:2      0   2215 Registered
00:90:83:36:82:f1 201.1.1.107      12    1   3 1:2      0   1248 Registered
```

Related Commands

- **show modem** on page 459
- **show modem <mac-address>** on page 467

show interface cable <c/s/i> modem summary

Shows an abbreviated cable modem summary status for the current cable interface. Information includes modem count and associated DOCSIS initialization states.

Mode

Any.

Syntax

show interface cable <c/s/i> **modem summary**

Example

```
cli:192.168.208.3:root# show interface cable 1/1/1 modem summary

row count: 13


Modem Status Summary
Modem State           Chan 1  Chan 2  Chan 3  Chan 4  All Chan
--------------------  ------  ------  ------  ------  --------
DhcpReqRcv                 0       4       0       0         4
RegBpiTek                  0       1       0       0         1
Registered                 0       7       0       0         7
Ranging                    0       1       0       0         1
--------------------  ------  ------  ------  ------  --------
Total                      0      13       0       0        13
```

Related Commands

- **show modem summary** on page 478

show interface cable <c/s/i> modulation-profile

Displays the modulation profiles currently configured on the specified cable interface.

Mode

Any.

Syntax

show interface cable <c/s/i> **modulation-profile**

Example

```
cli:172.16.19.10:root# show interface cable 1/1/1 modulation-profile

Mod Interval Mod   Pream  Pream Diff FEC   FEC CW Scram Max   Guard Last Scram
Idx Usage    Type  Offset Len        Error Len    Seed  Burst Time  CW   Enable
                                                      Size
--- -------- ----- ------ ----- ---- ----- ------ ----- ----- ----- ---- ------
  1 Request  QPSK      56    64  no       0     16   338     2     8  no    yes
  1 Initial  QPSK       0   128  no       5     34   338     0    48  no    yes
  1 Station  QPSK       0   128  no       5     34   338     0    48  no    yes
  1 Short    QAM16    272   144  no       6     78   338     7     8  yes   yes
  1 Long     QAM16    256   160  no       8    220   338     0     8  yes   yes
  2 Request  QPSK      56    64  no       0     16   338     2     8  no    yes
  2 Initial  QPSK       0   128  no       5     34   338     0    48  no    yes
  2 Station  QPSK       0   128  no       5     34   338     0    48  no    yes
  2 Short    QPSK      48    72  no       5     78   338    12     8  yes   yes
  2 Long     QPSK      40    80  no       8    220   338     0     8  yes   yes

row count: 10
cli:172.16.19.10:root#
```

show interface cable <c/s/i> pll-state

Displays the Phase Lock Loop (PLL) state for the specified cable interface.

Mode

Any.

Syntax

show interface cable <c/s/i> pll-state

Example

```
cli:172.16.19.10:root# show interface cable 1/1/1 pll-state
PLL State                              normal
PLL Value                                   0

cli:172.16.19.10:root#
```

show interface cable <c/s/i> proxy-arp

Use this command to verify whether proxy ARP is enabled or disabled on the current cable interface.

Mode

Any.

Syntax

show interface cable <c/s/i> **proxy-arp**

Example

```
cli:172.16.19.10:root# show interface cable 1/1/1 proxy-arp
Simple Proxy Arp:                    Disable

cli:172.16.19.10:root#
```

show interface cable <c/s/i> qos

Use this command to verify QoS configuration for a specified cable interface.

Mode

Any.

Syntax

show interface cable <c/s/i> **qos**

Example

```
cli:192.168.220.222:root# show interface cable 1/1/1 qos

QOS     Service  Max        Guaranteed Max        Max        Baseline Status
Profile Priority Upstream   Bandwidth  Downstream Upstream   Privacy
Index            Bandwidth             Bandwidth  TxBurst
------- -------- ---------- ---------- ---------- ---------- -------- -------
   1025        7          0          0          0          0        2        1
   1026        1      64000          0    1000000          0        2        1
```

Related Commands

- **show qos** on page 509

show interface cable <c/s/i> sid

Displays information by service identifier (SID) of each cable modem registered with the CMTS.

Mode

Any.

Syntax

show interface cable <c/s/i> **sid** {<sid-number> | **stats**}

Arguments

<sid-number>	Specify the SID for which you want to display information.
stats	Displays SID statistics.

Example

The following example shows all cable modem by SID, then uses the **stats** keyword to show statistics for each SID.

```
cli:192.168.208.3:root# show interface cable 1/1/1 sid

row count: 12

SID   QOS    Create Time       Class MAC Address       IP Address
                               ID
----- ------ ----------------- ----- ----------------- ---------------
    1   1026 01-08-29 18:39:14     1 00:10:95:04:0a:b7 201.1.1.109
    2   1026 01-08-29 18:42:03     1 00:90:96:00:39:f9 201.1.1.101
    3   1025 01-08-29 18:26:25   152 00:10:95:04:0a:c3 0.0.0.0
    4      0 01-08-29 18:42:05     0 00:90:83:36:82:ee 201.1.1.108
    5   1026 01-08-30 10:09:33     1 00:10:95:01:f0:05 201.1.1.103
    6   1026 01-08-29 18:40:47     1 00:10:95:04:0a:c4 201.1.1.110
    7   1026 01-08-29 18:42:03     1 00:90:96:00:29:6d 201.1.1.104
    8   1026 01-08-29 18:42:03     1 00:90:96:00:29:71 201.1.1.112
    9   1026 01-08-29 18:42:36     1 00:90:83:32:9f:8c 201.1.1.106
   10   1026 01-08-29 22:17:11     1 00:10:95:01:ef:d8 201.1.1.100
   11   1026 01-08-29 18:42:03     1 00:90:96:00:39:7f 201.1.1.102
   12   1026 01-08-29 18:41:30     1 00:90:83:36:82:f1 201.1.1.107

cli:192.168.208.3:root# show interface cable 1/1/1 sid stats

row count: 12

SID   In Pkts    In Disc    Out Pkts   Out Disc   Bw Reqs    Bw Grants
----- ---------- ---------- ---------- ---------- ---------- ----------
    1        422          0        271          0        483        483
    2        142          0          0          0        218        218
    3          0          0          0          0          0          0
    4        143          0          1          0        174        174
    5         63          0         14          0        474        474
    6        416          0        271          0        477        477
    7        142          0          0          0        230        230
    8        142          0          0          0        234        234
    9        230          0         87          0        254        254
   10        178          0         60          0        393        393
   11        142          0          0          0        214        214
   12        527          0        204          0        546        546
```

show interface cable <c/s/i> signal-quality

Use this command to display signal quality information for the specified cable interface.

Mode

Any.

Syntax

show interface cable <c/s/i> **signal-quality**

Example

```
cli:172.16.19.10:root# show interface cable 1/1/1 signal-quality
Cable 1/1/1                       Upstream 1
Contention Intervals                   False
Unerrored Codewords                        1
Correctable Codewords                      0
Uncorrectable Codewords                    0
Signal-to-Noise Ratio                   42.1  (dB)
Equalization Data

Cable 1/1/1                       Upstream 2
Contention Intervals                   False
Unerrored Codewords                   273146
Correctable Codewords                      0
Uncorrectable Codewords                    1
Signal-to-Noise Ratio                   25.9  (dB)
Equalization Data

Cable 1/1/1                       Upstream 3
Contention Intervals                   False
Unerrored Codewords                        0
Correctable Codewords                      0
Uncorrectable Codewords                    0
Signal-to-Noise Ratio                   42.1  (dB)
Equalization Data

Cable 1/1/1                       Upstream 4
Contention Intervals                   False
Unerrored Codewords                        0
Correctable Codewords                      0
Uncorrectable Codewords                    0
Signal-to-Noise Ratio                   42.1  (dB)
Equalization Data

        Timer: elapsed time: 15750 mSeconds

cli:172.16.19.10:root#
```

show interface cable <c/s/i> upstream

Displays upstream port information for the specified cable interface.

Mode

Any.

Syntax

show interface cable <c/s/i> **upstream** [<port-number>]

Arguments

<port-number>	Specify the SID for which you want to display information.

Example

```
cli:192.168.208.3:root# show interface cable 1/1/1 upstream
Upstream Channel Id        1 (1 / 1 / 3 / 2)
Center Frequency                    20.0  (MHz)
Channel Width                     3200.0  (KHz)
Slot Size                              2  (uSec)
Receive Power                          0  (TenthdBmV)
Voice BW Reservation                65.0  (%)
Modulation Profile                     1
Tx Timing Offset                       0
Tx Backoff Start                       5
Tx Backoff End                        10
Ranging Backoff Start                  2
Ranging Backoff End                    3

Admin Status                          up
Operational Status                    up

Stats:
In Octets                         328945
In Unicast Packets                    93
In Multicast Packets                   0
In Broadcast Packets                 703
In Errors                              0
In Discards                            0
--More--
```

show interface ethernet

Use this command to show statistics for a specified Ethernet interface.

Mode

Any.

Syntax

show interface ethernet <c/s/i> {**in-counters** | **out-counters**}

Arguments

in-counters	Displays incoming traffic statistics for the specified Ethernet interface.
out-counters	Displays outgoing traffic statistics for the specified Ethernet interface.

Example

The following example displays Ethernet in-counters statistics:

```
cli:172.16.19.10:root# show interface/ethernet 1/11/1 in-counters
Interface                          1 / 11 / 1
Type                                 ethernet
In octets                                   0
In unicast                                  0
In multicast                                0
In broadcast                                0
```

The fields in the in-counters display are as follows:

Field	Description
Interface	Chassis/Slot/Port of the specified interface.
Type	Interface type (Ethernet).
In octets	Total number of Octets that have been received on this interface, including framing characters.
In unicast	Number of Unicast packets that have been received on this interface.
In multicast	Number of Multicast packets that have been received on this interface.
in broadcast	Number of Broadcast packets that have been received on this interface.

If out-counters were displayed, the following fields would appear:

Field	Description
Interface	Chassis/Slot/Port of the specified interface.
Type	Interface type (Ethernet).
Out octets	Total number of octets that have been transmitted out of this interface, including framing characters.
Out unicast	Total number of Unicast packets that have been transmitted out of this interface.
Out multicast	Total number of Multicast packets that have been transmitted out of this interface.
Out broadcast	The total number of Broadcast packets that have been transmitted out of this interface.

Related Commands

- **show in-counters** on page 413
- **show out-counters** on page 491

show interface pos

Displays configuration parameters and statistics for a POS interface.

Mode

Any.

Syntax

show interface pos {***** | <c/s/i>}

Arguments

{***** | <c/s/i>}	Specifies * (display information on all POS interfaces) or <c/s/i> (display information on the specified interface only).

Example

```
cli:192.168.208.3:root# show interface pos 1/8/1
-------------------------------------------------------
Interface Type                          sonet
POS 1/8/1 (line protocol)              closed
-------------------------------------------------------
Hardware is Packet Over Sonet
Internet Address
Rx Giants                                  0
Bad FCS's                                  0
Bad Addresses                              0
Bad Controls                               0
Local MRU                               1518  (bytes)
Remote MRU                                 0  (bytes)
FCS Size                                  32  (bits)
Transmission Errors (Tx)                   0
Rx Abort                                   0
Rx Runts                                   0
Interface Type                           ppp
Interface Speed                       155520  (Kbits)
--More--
```

Related Commands

- **show interface** on page 414

show ip

Displays the contents of the routing table.

Mode

Any.

Syntax

show ip

Example

```
cli:192.135.241.21:root# show ip

row count: 13

Protocol Route  Destination     Net Mask        Next Hop        Metric C/S/I
         Type
-------- ------ -------------- -------------- -------------- ------ ------
Local    Local  5.5.5.0        255.255.255.0  5.5.5.2             0 1/9/8
Local    Local  12.2.2.0       255.255.255.0  12.2.2.1            0 1/2/1
Local    Local  12.2.4.0       255.255.255.0  12.2.4.1            0 1/4/1
Local    Local  14.2.4.0       255.255.255.0  14.2.4.1            0 1/4/1
Local    Local  172.16.30.0    255.255.255.0  172.16.30.1         0 1/4/1
Local    Local  172.16.31.0    255.255.255.0  172.16.31.1         0 1/4/1
Local    Local  172.16.32.0    255.255.255.0  172.16.32.1         0 1/4/1
Local    Local  172.102.91.0   255.255.255.0  172.102.91.1        0 1/9/1
Local    Local  172.102.92.0   255.255.255.0  172.102.92.1        0 1/9/2
Local    Local  172.102.93.0   255.255.255.0  172.102.93.1        0 1/9/3
Local    Local  172.102.94.0   255.255.255.0  172.102.94.1        0 1/9/4
Local    Local  192.102.102.0  255.255.255.0  192.102.102.1       0 1/2/1
Local    Local  192.102.104.0  255.255.255.0  192.102.104.1       0 1/4/1
```

Related Commands

- **ip route** on page 243
- **ip route default** on page 245

show ip {mroute | igmp}

Mode

```
root
interface:<type>:csi(<c/s/i>)
```

Syntax

```
show ip {mroute [<group-address> | summary] | igmp {groups
[<group-address> | <c/s/i>] | proxy | interface [<c/s/i>]}}
```

Arguments

mroute
[<group-address> | **summary**]

Specify one of these argument combinations:

- The **mroute** argument without a group address or **summary**. This action displays details on all multicast routes in the IP multicast forwarding table. For example:

  show ip mroute

- The **mroute** argument with a group address. This action displays only the multicast route associated with the group address. For example:

  show ip mroute 225.1.1.1

- The **mroute** argument with **summary**. This action displays summary information on all multicast routes. For example:

  show ip mroute summary

igmp groups [<group-address> | <c/s/i>]

Specify one of these argument combinations:

- The **igmp groups** arguments without a group address or an interface. This action displays details on all multicast groups that the Cuda 12000 has joined. For example:

  show ip igmp groups

- The **igmp groups** arguments with a group address. This action displays details on the specified multicast group only. For example:

  show ip igmp groups 225.2.3.1

- The **igmp groups** arguments with an interface. This action displays details on the multicast groups that the Cuda 12000 has joined on the specified interface only. For example:

  show ip igmp groups 1/3/2

igmp proxy

Displays all interfaces that act as IGMP proxies and the multicast groups for which they proxy. For example:

show ip igmp proxy

igmp interface [<c/s/i>]

Specify one of these argument combinations:

- The **igmp interface** arguments without an interface. This action displays all interfaces that can perform multicast forwarding. For example:

  show ip igmp interface

- The **igmp interface** arguments with an interface that can perform multicast forwarding. This action displays information on the specified interface only. For example:

  show ip igmp interface 1/11/5

Example 1

In this example, the user displays all interfaces running IGMP, and then displays IGMP details on interface 1/1/1.

An IP address of 0.0.0.0 means that the interface functions as an IGMP host.

An IP address other than 0.0.0.0 (such as 201.0.0.1) means that the interface functions as an IGMP router. This address identifies the querier on this network.

```
cli:192.168.220.208:root# show ip igmp interface

row count: 9

Querier         Interface Version Query    Max Resp Joins Wrong   Groups
                                  Interval Time           Queries
--------------- --------- ------- -------- -------- ----- ------- ------
201.0.0.1          1/1/1       2      125       10     4       0      2
0.0.0.0           1/11/1       2      125       10     1       0      1
0.0.0.0           1/11/2       2      125       10     1       0      1
0.0.0.0           1/11/3       2      125       10     0       0      0
0.0.0.0           1/11/4       2      125       10     0       0      0
0.0.0.0           1/11/5       2      125       10     0       0      0
0.0.0.0           1/11/6       2      125       10     0       0      0
0.0.0.0           1/11/7       2      125       10     0       0      0
0.0.0.0           1/11/8       2      125       10     0       0      0
cli:192.168.220.208:root# show ip igmp interface 1/1/1
IGMP Group Interface Parameters on 1/1/1
Multicast forwarding                Enabled

IP Address                        201.0.0.1

Interface is                   IGMP Querier
Interface                             1/1/1
Querier                         201.0.0.1
Up Time                          11:42:50
Version                                 2
Query Interval                        125
Max Resp Time                          10
Robustness                              2
Joins                                   4
Wrong Queries                           0
Groups                                  2
Last Member Interval                    1
Version 1 Querier Timer          00:00:00
```

Example 2

In this example, the user displays all IGMP proxies.

Status of the proxy can be active or backup. Active status means that the proxy is currently in use. Backup status means that the proxy is currently not in use.

For descriptions of Group Address, Mask, Interface, and Metric, see the description of the **ip igmp proxy** command.

```
cli:192.168.220.230:root# show ip igmp proxy

row count: 2

Group Address   Mask            Interface       Metric Status
--------------- --------------- --------------- ------ ------------
225.4.0.0       255.255.0.0              1/1/1      1       active
225.5.0.0       255.255.0.0              1/1/1      1       active

cli:192.168.220.230:root#
```

Related Commands

- **ip igmp** on page 187
- **ip igmp proxy** on page 189

show ip address

Displays the IP interfaces configured on the current interface. Note that you configure IP interfaces using the **ip address** command.

Mode

```
interface:<type>:csi(<c/s/i>)
```

Syntax

```
show ip address
```

Example

```
cli:192.168.208.3:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:192.168.208.3:interface:cable:csi(1/1/1)# show ip address
Chassis/Slot/Interface      1/1/1

IP Address       Net Mask         Interface   Priority
---------------- ---------------- ---------- ----------
   192.168.19.50    255.255.255.0    8781825    Primary

row count: 1
cli:192.168.208.3:interface:cable:csi(1/1/1)# ip address 192.167.30.3
255.255.255.0
cli:192.168.208.3:interface:cable:csi(1/1/1)# show ip address
Chassis/Slot/Interface      1/1/1

IP Address       Net Mask         Interface   Priority
---------------- ---------------- ---------- ----------
    192.167.30.3    255.255.255.0    8781825    Primary
   192.168.19.50    255.255.255.0    8781825      Other

row count: 2
cli#
```

Related Commands

- **ip address** on page 183
- **show ip interface** on page 445

show ip filter

Use this command to show whether packet filtering on a selected cable interface is enabled or disabled.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
show ip filter
```

Example

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# show ip filter

IP Filter In                            enable
IP Filter Out                          disable

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **access-class** on page 45
- **access-list** on page 47
- **ip filter** on page 185

show ip interface

Displays the IP interfaces configured throughout the system, or for a specific physical interface. Note that you configure IP interfaces using the **ip address** command.

Mode

To display a list of IP addresses added to the specified network interface:

`interface:<type>:csi(<c/s/i>)`

To display a list of IP addresses configured on all network interfaces within the system:

`root`

Syntax

`show ip interface`

Example

```
cli:192.168.220.207:interface:cable:csi(1/1/1)# show ip interface
Chassis/Slot/Interface     1 / 1 / 1 (8781825)
Description                CATV MAC: Broadcom BCM3210
Admin Status                            up
Oper Status                             up
Mtu                                   1500  (bytes)

IP Address                        201.0.0.1
Net Mask                     255.255.255.0
Interface                          8781825
Priority                           Primary

RIP configuration:

OSPF configuration:
Area ID                            1.1.1.1
Type                                 Bcast
Priority                                 5
Transit Delay                            1
Retrans Int                              5
Hello Int                               10
Dead Int                                40
Poll Int                                 0
Admin Stat                         Enabled
Status                              Active
Auth Type                             None

Auth Key Id                              0
Cost                                     1
```

Related Commands

- **ip address** on page 183
- **show ip address** on page 443

show ip interface source-route

Displays the IP source routes configured on the current interface. Source routing allows you to configure the next-hop destination to which the interface forwards packets that do not match a local or default route in the routing table.

Mode

```
interface:<type>:csi(<c/s/i>)
```

Syntax

```
show ip interface source-route
```

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# show ip interface source-route

row count: 1

Address          Mask             NextHop
---------------- ---------------- ----------------
209.240.193.0    255.255.255.0    209.240.172.3

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **ip source-route** on page 246

show ip ospf

Displays OSPF interface parameters.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(ip-address)#
```

Syntax

show ip ospf

Example

```
cli:192.168.220.207:interface:cable:csi(1/1/1)# ip address 201.0.0.1
255.255.255.0
cli:192.168.220.207:interface:cable:csi(1/1/1):ip-address(201.0.0.1)# show
ip ospf
IP Address                          201.0.0.1
Area ID                               1.1.1.1
Type                                    Bcast
Priority                                    5
Transit Delay                               1
Retrans Int                                 5
Hello Int                                  10
Dead Int                                   40
Poll Int                                    0
Admin Stat                            Enabled
Status                                 Active
Auth Type                                None

Auth Key Id                                 0
Cost                                        1
```

Related Commands

- **ip ospf area-id** on page 191
- **ip ospf authentication** on page 193
- **ip ospf cost** on page 196
- **ip ospf dead-interval** on page 199
- **ip ospf hello-interval** on page 202
- **ip ospf priority** on page 205
- **ip ospf retransmit-interval** on page 208
- **ip ospf transit-delay** on page 211
- **router ospf** on page 359
- **show ip ospf** on page 448

show ip rip

Displays RIP configuration parameters for an IP interface.

Mode

```
interface:<type>:csi(<c/s/i>):ip-address(<ip-address>)#
```

Syntax

show ip rip

Example

```
cli:interface:ethernet:csi(1/11/1):ip-address(192.168.16.3)# show ip rip
IP Address                      192.168.16.3
Send Version                RIPv1 Compatible
Receive Version               RIPv1 Or RIPv2
Cost                                       1
Authentication ON                      False
Authentication Type        No Authentication
Authentication Key Id                      0
Send Default Only                      False
Send Default Also                      False
Default Cost                               0
Accept Default                          True
Accept Host Route                       True
Split Horizon                           True
Poisoned Reverse                        True
Status                                Active
```

Related Commands

- **ip rip accept default-route** on page 214
- **ip rip accept host-route** on page 216
- **ip rip authentication** on page 218
- **ip rip cost** on page 221
- **ip rip default cost** on page 223
- **ip rip disable** on page 225
- **ip rip enable** on page 227
- **ip rip neighbor** on page 229
- **ip rip poisoned-reverse** on page 231
- **ip rip receive-version** on page 233
- **ip rip send-version** on page 235
- **ip rip send default-also** on page 237
- **ip rip send default-only** on page 239
- **ip rip split-horizon** on page 241

show link-trap

Displays the status (Enabled or Disabled) of link up and link down traps for an interface.

Mode

```
interface:<type>:<c/s/i>
```

Syntax

```
show link-trap
```

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show link-trap
Link Up/Down Trap Enable             Enabled
```

Related Commands

- **link-trap** on page 249

show lookup

Displays the current Jini lookup service status.

Mode

`root`

Syntax

`show lookup`

Example

```
cli:192.168.208.3:root# show lookup
# JINI lookup service (reggie) is stopped.
```

Related Commands

- **lookup** on page 250

show mac

Displays Media Access Control (MAC) configuration and statistics for the current CMTS interface.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

show mac

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show mac
Insertion Interval                        10  (centisec)
Invited Ranging Attempts                  16
Sync Interval                              5  (millisec)
UCD Interval                            2000  (millisec)

Hardware Map Timer                      2000  (microsec)
Periodic Ranging Interval                 15  (secs)
Plant Delay                             1600

PLL State                             normal
PLL Value                                  0

Stats:
Admin Status                              up
Operational Status                        up
In Octets                          112923412
In Unicast Packets                   3308993
In Multicast Packets                       0
In Broadcast Packets                       1
In Errors                               2396
In Discards                                0
Out Octets                        3794178837
Out Unicast Packets                  3428781
Out Multicast Packets              466970710
Out Broadcast Packets                  19996
Out Errors                                 0
Out Discards                               0

Invalid Range Requests                     0
Ranging Aborts                             0
```

```
Invalid Registrations                    0
Failed Registrations                     0
Invalid Data Requests                    0
T5 Timeouts                              0
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421

show map-list

Displays the map list information used for importing and exporting routes.

Mode

To display import OSPF map list information:

```
router:ospf:import
```

To display export OSPF map list information:

```
router:ospf:export
```

To display import RIP map list information:

```
router:rip:import
```

To display export RIP map list information:

```
router:rip:export
```

Syntax

```
show map-list
```

Example

```
cli:172.16.19.10:root# router ospf export
mode: router:ospf:export
cli:172.16.19.10:router:ospf:export# show map-list

Filter  Template Status
        Count
------- -------- --------------
      1        1 Not In Service

row count: 1
cli:172.16.19.10:router:ospf:export#
```

Related Commands

- **map-list** on page 251
- **match** on page 254
- **override** on page 309
- **route-map** on page 354
- **router ospf** on page 359

show mode

Displays the current mode. Use this command to verify the current mode that you are in while using the CLI.

Mode

Any.

Syntax

show mode

Example

```
cli:172.16.19.10:router:ospf:export# show mode
mode: router:ospf:export
cli:172.16.19.10:router:ospf:export#
```

show modem

Displays cable modem status for DOCSIS 1.0 and DOCSIS 1.1 cable modems and QoS definitions for DOCSIS 1.1 cable modems.

Mode

To display cable modem status per interface and per cable modem:

```
root
```

To display cable modem status per interface and per cable modem:

```
interface:cable:csi(<c/s/i>)
```

To display the QoS definition for DOCSIS 1.1 cable modems:

```
interface:cable:csi(<c/s/i>)
```

Syntax

Within root or interface cable <c/s/i> mode:

```
show [interface cable <c/s/i>] modem [<mac-address> <arguments>]
```

Within interface cable <c/s/i> mode:

```
show [interface cable <c/s/i>] modem [<mac-address> <arguments>]
  [{cm-filter | cpe-control | cpe-hosts | stats | counts |
  upstream <channel-id> | service-flow [<sfid>] {classifier
  [<class-id>] | log [<log-id>] | parameter-set [<type>] | stats |
  upstream}}]
```

Arguments

interface cable <c/s/i>	Within root or interface cable <c/s/i> mode. Specifies the chassis, slot, interface of the cable modem for which you want to display the status.
<mac-address> <arguments>	Within root or interface cable <c/s/i> mode. Specifies the MAC address for this cable modem, followed by additional arguments. See the description of **show modem <mac-address>** on page 467 for more information.
cm-filter	Within interface cable <c/s/i> mode. Displays, for each cable modem, the cable modem upstream and downstream filter group assignments and the CPE upstream and downstream filter group assignments.
cpe-control	Within interface cable <c/s/i> mode. Displays the subscriber management parameters for each cable modem.
cpe-hosts	Within interface cable <c/s/i> mode. Displays the CPE devices for each cable modem. No CPE devices appear if the CMTS cannot learn IP addresses for the specified cable modem.
stats	Within interface cable <c/s/i> mode. Displays upstream and downstream byte and packet counts for each cable modem.
counts	Within interface cable <c/s/i> mode. Displays network statistics for each cable modem.
upstream <channel-id>	Within interface cable <c/s/i> mode. Displays cable modems on the specified upstream channel.

service-flow [<sfid>] {**classifier** [<class-id>] | **log** [<log-id>] | **parameter-set** [<type>] | **stats** | **upstream**}

Within interface cable <c/s/i> mode. Displays the 32-bit Service Flow Identifier (SFID) assigned by the CMTS at registration when issued without the <sfid> argument. When issued with the <sfid> argument, displays details on the specified service flow.

You may combine the service-flow argument with one of the following additional arguments:

- classifier [<class-id>] – Displays the classifier to which incoming packets are matched, when issued without the <class-id> argument. Optionally, you can display details on a specific classifier ID.
- log [<log-id>] – Indicates the unique identifier for a logged Service Flow, when issued with no <log-id> argument. Optionally, you can display details on a specific log ID.
- parameter-set [<type>] – Displays the type and QoS parameter characteristics of the Service Flow assigned to this interface. Valid Service Flow types are 1 (active), 2 (admitted), and 3 (provisioned).
- stats – Displays statistics for this Service Flow.
- upstream – Displays only upstream statistics for this Service Flow.

Example 1

The following is an example of **show modem** in root mode. The output is a display of the status of all cable modems. Except for the slot number, the same output is displayed by **show modem** in interface:cable:csi(<c/s/i>) mode.

```
cli:192.168.220.230:root# show modem

row count: 4

S  MAC Address       IP Address      SID  CID  CPE D:U Power  Timing Modem
l                                                  (dbMV)        State
o
t
-- ----------------- --------------- ---- ---- --- --- ------ ------ --------
 1 00:20:40:c0:82:08 201.0.0.102      538    1   0 1:1      0   1904 Registered
 1 00:20:40:6f:88:7e 201.0.0.100      540    1   0 1:1      0   1907 Registered
 1 00:10:18:00:20:65 201.0.0.103      539    1   0 1:1      0   1126 Registered
 1 00:a0:c5:23:f6:6c 201.0.0.101      536    1   0 1:1      0   1906 Registered
```

Example 2

The following is an example of **show** [**interface cable 1/1/1**] **modem <mac>** in root mode. The output is a display of the status of a single cable modem. Note that the same output is displayed by **show** [**interface cable 1/1/1**] **modem <mac>** in interface cable <c/s/i> mode.

```
cli:192.168.208.3:root# show interface cable 1/1/1 modem 00:90:83:36:82:f1
S l o t                                  1
MAC Address                00:90:83:36:82:f1
IP Address                       201.1.1.107
SID                                     12
CID                                      1
CPE                                      0
D:U                                    1:2
Power                                    0  (dbMV)
Timing                                1240
Modem State                       Registered
```

Example 3

The following is an example of a summary of the Service Flows for the specified interface.

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem service-flow

row count: 2

Mac Address       Service  Direction  Primary   Time Created   Class Name Sche
                  Flow ID                                                 Type

----------------- -------- ---------- --------- -------------- ---------- ---

00:90:83:36:82:ee      25 DS         True      01-08-08 20:29            undefi
00:90:83:36:82:ee      26 US         True      01-08-08 20:29            best e
```

Example 4

The following is an example of a summary of a specified Service Flow:

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem service-flow 26
SFID:                  26


row count: 1

Mac Address       Direction  Primary   Time Created   Class Name Scheduling
                                                                 Type
----------------- ---------- --------- -------------- ---------- -----------
00:90:83:36:82:ee US         True      01-08-08 20:29             best effort
```

Example 5

The following is an example of the statistics for a specified Service Flow on an upstream channel:

```
cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:EE service-flow 1
upstream
Cable Modem:               00:90:83:36:82:EE
SFID                                       1
Fragments                                  0
Discarded Fragments                        0
Concatenated Bursts                        0
cli:192.168.208.3:interface:cable:csi(1/1/1)#
```

Example 6

The following is a summary of the QoS parameter sets for a specified Service Flow:

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:ee
service-flow 26 parameter-set
Cable Modem:           00:90:83:36:82:ee
SFID:                  26


row count: 3

Param Type     Service  Max Traffic Min Reserved Max Concat Scheduling
               Class    Rate        Rate         Burst      Type
               Name     (bits/sec)  (bits/sec)   (bytes)
-------------- -------- ----------- ------------ ---------- ------------
active(1)                       0            0          0 best effort
admitted(2)                     0            0          0 best effort
provisioned(3)                  0            0          0 best effort
```

Example 7

The following is a display of an Admitted parameter set for a specified Service Flow:

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:ee
service-flow 26 parameter-set 2
Cable Modem:           00:90:83:36:82:ee
SFID:                  26
Param Type                      admitted(2)
Service Class Name
Priority                                 0
Max Traffic Rate                         0  (bits/sec)
Max Traffic Burst                     1522  (bytes)
Min Reserved Rate                        0  (bits/sec)
Min Reserved Packet                     64  (bytes)
Active Timeout                           0  (secs)
Admitted Timeout                       200  (secs)
Max Concat Burst                         0  (bytes)
Scheduling Type                 best effort
Nominal Polling Interval                 0  (usecs)
Tolerable Poll Jitter                    0  (usecs)
Unsolicited Grant Size                   0  (bytes)
Nominal Grant Interval                   0  (usecs)
Tolerable Grant Jitter                   0  (usecs)
Grants Per Interval                      0
TOS AND Mask              11111111
TOS OR Mask               00000000
Max Latency                              0  (usecs)
Request Policy Octets     00:00:00:00
Bit Map
    trafficPriority                   off
    maxTrafficRate                    off
    maxTrafficBurst                   off
    minReservedRate                   off
    minReservedPkt                    off
    activeTimeout                     off
    admittedTimeout                   off
    maxConcatBurst                    off
    schedulingType                    off
    requestPolicy                     off
    nomPollInterval                   off
    tolPollJitter                     off
    unsolicitGrantSize                off
    nomGrantInterval                  off
    tolGrantJitter                    off
    grantsPerInterval                 off
    tosOverwrite                      off
```

Example 8

The following example displays the subscriber management parameters for all cable modems on an interface:

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem cpe-control

row count: 12
MAC Address       IP Address      MAX IP     Active     Learnable
----------------- --------------- ---------- ---------- ----------
00:10:95:04:0a:b7 201.1.1.109     16         True       True
00:90:96:00:39:f9 201.1.1.101     16         True       True
00:10:95:04:0a:c3 0.0.0.0         16         True       True
00:90:83:36:82:ee 201.1.1.108     16         True       True
00:10:95:01:f0:05 201.1.1.103     16         True       True
00:10:95:04:0a:c4 201.1.1.110     16         True       True
00:90:96:00:29:6d 201.1.1.104     16         True       True
00:90:96:00:29:71 201.1.1.112     16         True       True
00:90:83:32:9f:8c 201.1.1.106     16         True       True
00:10:95:01:ef:d8 201.1.1.100     16         True       True
00:90:96:00:39:7f 201.1.1.102     16         True       True
00:90:83:36:82:f1 201.1.1.107     16         True       True
```

Example 9

The following example displays a CPE device that uses the cable modem with the MAC address of 00:90:83:36:82:f1. Note that "Learned" indicates that the CMTS discovered the CPE IP address through upstream packets that were sent by the CPE.

```
cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:f1 cpe-hosts
MAC Address                00:90:83:36:82:f1
IP Address                       201.1.1.111

row count: 1

CPE IP Address   CPE MAC Address   Source
---------------- ----------------- -----------
201.1.2.100      00:b0:d0:72:b2:93     Learned
```

Related Commands

- **show interface cable <c/s/i> modem** on page 423
- **show modem <mac-address>** on page 467
- **show modem summary** on page 478

show modem <mac-address>

Queries a specified cable modem directly to retrieve and display its current MIB values, which are described in MIB II, RFC2669, RFC2670, the Baseline Privacy Interface (BPI) and BPI Plus specifications, and other cable modem networking standards documents and MIBs. If you do not specify a read community string, the command sends the default read community string to the cable modem.

Mode

```
root
interface:cable:csi(<c/s/i>)
```

Syntax

```
show [interface cable <c/s/i>] modem <mac-address>
  {cm {bpi-base | bpi-plus {base | tek} | bpi-tek | device [event
  {config | control | list} [read-community <string>] | downstream |
  interface | mac | qosprofile | service | signal-quality | status |
  system | upstream} | cm-filter | cpe-control | cpe-hosts |
  service-flow {classifier [<cid> | phs] | log [<log-id>] |
  parameter-set [<type>] | stats | upstream}}
```

Arguments

interface cable <c/s/i>	Specifies the chassis, slot, interface of the cable modem for which you want to display the status.
<mac-address>	Specifies the MAC address of the cable modem.

cm {**bpi-base** | **bpi-plus** {**base** | **tek**} | **bpi-tek** | **device** [**event** {**config** | **control** | **list**}] | **downstream** | **interface** | **mac** | **qosprofile** | **service** | **signal-quality** | **status** | **system** | **upstream**} [**read-community** <string>]	Specifies one of the following values: ■ bpi-base – Displays BPI base information for the cable modem. ■ bpi-plus {base | tek} – Displays BPI plus information for the cable modem. The base argument displays BPI plus base information. The tek argument displays BPI plus traffic encryption key (TEK) information. ■ bpi-tek – Displays BPI TEK information for the cable modem. ■ device [event {config | control | list}] – Displays device status and control information for the cable modem. Optionally, you can specify the event argument, which displays event configuration information, event control information or a list of events that the cable modem has generated. ■ downstream – Displays downstream channel information for the specified cable modem. ■ interface – Displays information on the CPE Ethernet interface, MAC interface, downstream interface, upstream interface, and diagnostic Ethernet interface. ■ mac – Displays the attributes of the specified cable modem's MAC interface. ■ qosprofile – Displays the attributes of the specified cable modem's QoS profile. ■ service – Displays upstream service queue attributes for the specified cable modem. ■ signal-quality – Displays the physical layer signal quality of downstream channels for the specified cable modem. ■ status – Displays several status fields and counters for the specified cable modem. ■ system – Displays standard MIB II system group objects for the specified cable modem (contact, system name, and location). ■ upstream – Displays upstream channel information for the specified cable modem. Optionally, you can follow each of these arguments with the **read-community** keyword followed by a community string.
cm-filter	Displays upstream and downstream filters for cable modem traffic and for CPE traffic that the specified cable modem processes.
cpe-control	Displays subscriber management parameters for CPE associated with the specified cable modem.

cpe-hosts

Displays the CPE devices associated with each cable modem. No CPE devices appear if the CMTS cannot discover IP addresses for the specified cable modem.

service-flow {**classifier** [<cid> | **phs**] | **log** [<log-id>] | **parameter-set** [<type>] | **stats** | **upstream**}

Displays details on the DOCSIS 1.1 Service Flow that the CMTS assigns to the specified cable modem at registration.

You may combine the service-flow argument with one of the following additional arguments:

- classifier [<cid> | **phs**] – When issued without any arguments, displays information on the packet classifier and payload header suppression (PHS) filter against which packets are matched. Optionally, you can display details on a specific classifier ID (with the <cid> variable) or PHS filter (with the **phs** keyword).
- log [<log-id>] – Indicates the unique identifier for a logged Service Flow, when issued with no <log-id> argument. Optionally, you can display details on a specific log ID.
- parameter-set [<type>] – Displays the type and QoS parameter characteristics of the Service Flow assigned to this interface. Valid Service Flow types are 1 (active), 2 (admitted), and 3 (provisioned).
- stats – Displays statistics for this Service Flow.
- upstream – Displays only upstream statistics for this Service Flow.

Example 1

```
cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:ee cm bpi-plus base
S l o t                                  1
MAC Address                 00:90:83:36:82:ee
IP Address                        201.1.1.108

docsBpi2CmPrivacyEnable                 True
docsBpi2CmPublicKey
30:81:89:02:81:81:00:f8:81:8a:66:ce:2c:26:87:e6:6e:cf


:5e:4d:28:b6:5c:fd:33:64:60:6f:3f:5e:f0:57:09:25:7c:8


4:6e:15:01:02:db:4c:13:67:0b:43:15:cc:89:cc:17:f4:59:


f6:a0:ae:56:a0:72:58:f2:97:6c:77:86:0b:98:04:43:0a:f3


:08:23:32:61:ea:54:83:9e:e7:11:2c:79:d4:f1:de:0a:05:4


7:82:3e:72:d7:47:52:4a:0c:55:f0:1a:8b:80:a0:13:fa:fe:


1a:ae:f3:03:af:30:ee:43:97:d9:30:c3:31:ef:35:0d:6e:ea

                          :b5:dc:8b:43:8c:34:78:82:93:52:4d:02:03:01:00:01
docsBpi2CmAuthState               Authorized
docsBpi2CmAuthKeySequenceN                 0
docsBpi2CmAuthExpiresOld   2001-1-1,0:0:0.0,430:0
docsBpi2CmAuthExpiresNew   2001-1-8,0:0:9.0,430:0
docsBpi2CmAuthReset                    False
docsBpi2CmAuthGraceTime                  100
docsBpi2CmTEKGraceTime                    61
docsBpi2CmAuthWaitTimeout                 30
docsBpi2CmReauthWaitTimeou                30
docsBpi2CmOpWaitTimeout                   10
docsBpi2CmRekeyWaitTimeout                10
docsBpi2CmAuthRejectWaitTi                60
docsBpi2CmSAMapWaitTimeout                10
docsBpi2CmSAMapMaxRetries                 10
docsBpi2CmAuthentInfos                     1
docsBpi2CmAuthRequests                     1
docsBpi2CmAuthReplies                      1
```

```
docsBpi2CmAuthRejects                       0
docsBpi2CmAuthInvalids                      0
docsBpi2CmAuthRejectErrorC                none
docsBpi2CmAuthRejectErrorS
docsBpi2CmAuthInvalidError                none
docsBpi2CmAuthInvalidError
```

Example 2

```
cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:ee cm device event
config
S l o t                                     1
MAC Address                  00:90:83:36:82:ee
IP Address                         201.1.1.108

Syslog Server                        201.1.1.1

Threshold                                   0
Interval                                    1
Admin Status                     unconstrained
Throttle Inhibited                       False
```

Example 3

```
cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:ee cm downstream
S l o t                                     1
MAC Address                  00:90:83:36:82:ee
IP Address                         201.1.1.108

docsIfDownChannelId                         1
docsIfDownChannelFrequency          507000000
docsIfDownChannelWidth                6000000
docsIfDownChannelModulatio                  3
docsIfDownChannelInterleav                  5
docsIfDownChannelPower                     24
docsIfDownChannelAnnex      xxxxxxxxxxxxxx
```

Example 4

```
cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:ee cm interface
S l o t                                  1
MAC Address                 00:90:83:36:82:ee
IP Address                        201.1.1.108

Description                 Ethernet CPE Interface
Type                                ethernet
Admin Status                              up
Oper Status                          dormant
In Octets                                  0
In Unicast Packets                         0
In Multicast Packets                       0
In Broadcast Packets                       0
In Errors                                  0
In Discards                                0
Out Octets                                 0
Out Unicast Packets                        0
Out Multicast Packets                      0
Out Broadcast Packets                      0
Out Errors                                 0
Out Discards                            1849
Description                 RF MAC Interface
Type                              docsCableMac
Admin Status                              up
Oper Status                               up
In Octets                              134720
In Unicast Packets                         64
In Multicast Packets                      692
In Broadcast Packets                     1158
In Errors                                  0
In Discards                                0
Out Octets                             30572
Out Unicast Packets                      205
Out Multicast Packets                      0
Out Broadcast Packets                      4
Out Errors                                 0
Out Discards                               0
Description                 RF Downstream Interface
Type                                docsCableDS
Admin Status                              up
Oper Status                               up
In Octets                       2609715265
In Unicast Packets                  304511
In Multicast Packets               61114186
In Broadcast Packets                   1158
```

```
In Errors                                  0
In Discards                            12178
Out Octets                                 0
Out Unicast Packets                        0
Out Multicast Packets                      0
Out Broadcast Packets                      0
Out Errors                                 0
Out Discards                               0
Description                RF Upstream Interface
Type                             docsCableUS
Admin Status                              up
Oper Status                               up
In Octets                                  0
In Unicast Packets                         0
In Multicast Packets                       0
In Broadcast Packets                       0
In Errors                                  0
In Discards                                0
Out Octets                            242806
Out Unicast Packets                     6468
Out Multicast Packets                      0
Out Broadcast Packets                      4
Out Errors                                 0
Out Discards                               0
```

Example 5

```
cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:ee cm mac
S l o t                                    1
MAC Address                00:90:83:36:82:ee
IP Address                       201.1.1.108

docsIfCmCmtsAddress        00:50:57:00:18:e8
docsIfCmCapabilities       Packet Transmission
docsIfCmRangingRespTimeout xxxxxxxxxxxxxx
docsIfCmRangingTimeout                    20
```

Example 6

```
cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:ee cm qosprofile
S l o t                                 1
MAC Address                00:90:83:36:82:ee
IP Address                       201.1.1.108

docsIfQosProfPriority                      0
docsIfQosProfMaxUpBandwidt                 0
docsIfQosProfGuarUpBandwid                 0
docsIfQosProfMaxDownBandwi                 0
docsIfQosProfMaxTxBurst                 1522
docsIfQosProfBaselinePriva              True
docsIfQosProfStatus                   Active
```

Example 7

```
cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:ee cm service
S l o t                                 1
MAC Address                00:90:83:36:82:ee
IP Address                       201.1.1.108

docsIfCmServiceQosProfile                  1
docsIfCmServiceTxSlotsImme                 0
docsIfCmServiceTxSlotsDed               3025
docsIfCmServiceTxRetries                   0
docsIfCmServiceTxExceededs                 0
docsIfCmServiceRqRetries                   0
docsIfCmServiceRqExceededs                 0
```

Example 8

```
cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:ee cm
signal-quality
S l o t                                 1
MAC Address                00:90:83:36:82:ee
IP Address                       201.1.1.108

docsIfSigQIncludesContenti                 2
docsIfSigQUnerroreds              2748995915
docsIfSigQCorrecteds                       0
docsIfSigQUncorrectables                   0
docsIfSigQSignalNoise                    348
docsIfSigQMicroreflections                21
docsIfSigQEqualizationData
```

Example 9

```
cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:ee cm upstream
S l o t                                 1
MAC Address                00:90:83:36:82:ee
IP Address                       201.1.1.108

docsIfUpChannelId                       2
docsIfUpChannelFrequency         23200000
docsIfUpChannelWidth              3200000
docsIfUpChannelModulationP              0
docsIfUpChannelSlotSize                 2
docsIfUpChannelTxTimingOff          11671
docsIfUpChannelRangingBack              2
docsIfUpChannelRangingBack              3
docsIfUpChannelTxBackoffSt              5
docsIfUpChannelTxBackoffEn             10
```

Example 10

```
cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:ee cm-filter
S l o t                                 1
MAC Address                00:90:83:36:82:ee
IP Address                       201.1.1.108

CPE DS Filter Group                     0
CPE US Filter Group                    10
CM DS Filter Group                     10
CM US Filter Group                      0
```

Example 11

```
cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:ee cpe-control
S l o t                                 1
MAC Address                00:90:83:36:82:ee
IP Address                       201.1.1.108

MAX IP                                 16
Active                               True
Learnable                            True
```

Example 12

```
cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:ee service-flow
Cable Modem:           00:90:83:36:82:ee

row count: 2

Service  Direction  Primary   Time Created   Class Name Scheduling
Flow ID                                                Type
-------- ---------- --------- -------------- ---------- -----------
      33 DS         True      01-08-04 15:21            undefined
      34 US         True      01-08-04 15:21            best effort

cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:ee service-flow 33
Cable Modem:           00:90:83:36:82:ee
SFID:                  33


row count: 1

Direction  Primary   Time Created   Class Name Scheduling
                                              Type
---------- --------- -------------- ---------- -----------
DS         True      01-08-04 15:21            undefined

cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:ee service-flow 33
parameter-set 1
Cable Modem:           00:90:83:36:82:ee
SFID:                  33

Param Type                          active(1)
Service Class Name
Priority                                   0
Max Traffic Rate                           0  (bits/sec)
Max Traffic Burst                       1522  (bytes)
Min Reserved Rate                          0  (bits/sec)
Min Reserved Packet                       64  (bytes)
Active Timeout                             0  (secs)
Admitted Timeout                         200  (secs)
Max Concat Burst                           0  (bytes)
Scheduling Type                    undefined
Nominal Polling Interval                   0  (usecs)
Tolerable Poll Jitter                      0  (usecs)
Unsolicited Grant Size                     0  (bytes)
Nominal Grant Interval                     0  (usecs)
Tolerable Grant Jitter                     0  (usecs)
Grants Per Interval                        0
```

```
TOS AND Mask                11111111
TOS OR Mask                 00000000
Max Latency                            0  (usecs)
Request Policy Octets       00:00:00:00
Bit Map

    trafficPriority                    off
    maxTrafficRate                     off
    maxTrafficBurst                    off
    minReservedRate                    off
    minReservedPkt                     off
    activeTimeout                      off
    admittedTimeout                    off
    maxConcatBurst                     off
    schedulingType                     off
    requestPolicy                      off
    nomPollInterval                    off
    tolPollJitter                      off
    unsolicitGrantSize                 off
    nomGrantInterval                   off
    tolGrantJitter                     off
    grantsPerInterval                  off
    tosOverwrite                       off
    maxLatency                         off
cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:ee service-flow
stats
Cable Modem:           00:90:83:36:82:ee

row count: 2

SFID Dir Primary Packets Octets Time Created   Time     Policed     Policed
                                               Active   Drop        Delay
                                                        Packets     Packets
---- --- ------- ------- ------ -------------- -------- ----------- ---------
  33  DS    True       3    948 01-08-04 15:21 01:19:50          0           0
  34  US    True      10   1460 01-08-04 15:21 01:19:50          0           0
```

Related Commands

- **show interface cable <c/s/i> modem** on page 423
- **show modem** on page 459
- **show modem summary** on page 478

show modem summary

Shows an abbreviated cable modem summary status for the current cable interface. Information includes modem count and associated DOCSIS initialization states.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

show modem summary

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem summary

row count: 12


Modem Status Summary
Modem State           Chan 1  Chan 2  Chan 3  Chan 4  All Chan
--------------------  ------  ------  ------  ------  --------
RegBpiTek                  0       1       0       0         1
Registered                 0      10       0       0        10
Ranging                    0       1       0       0         1
--------------------  ------  ------  ------  ------  --------
Total                      0      12       0       0        12
```

Related Commands

- **show interface cable <c/s/i> modem summary** on page 424

show mta <mac-address> base

Displays base or summary information about the MTA.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
show mta <mac-address> base [read-community <string>]
```

Arguments

<mac-address>	Specifies the MAC address of the MTA in xx:xx:xx:xx:xx:xx format.
read-community <string>	Specifies an optional community string for read access to the MTA base information.

Example

```
cli:interface:cable:csi(1/1/1)# show mta 00:20:40:c0:4a:ba base

S l o t                                  1
MAC Address                 00:20:40:c0:4a:ba
IP Address                        172.16.1.33
SID                                     115
CID                                       0
CPE                                       0
D:U                                     1:1
Power                                     0  (dbMV)
Timing                                ******
Modem State                     Registered

Serial Number
Hardware Version            Model: GW2E_ENG_L04.6.03_071301 BSP: 1.2/0
MAC Address                 00:ff:ff:c0:4a:ba
Fully Qual Domain Name      pp.default.net
EndPoint Count                            2
Voice Enabled                          True
Type ID                     pktc1.0:
Prov State                             pass
HTTP Access                 xxxxxxxxxxxxxx
```

The command output is divided into two sections. The top section displays information that you can display using the show modem command. The bottom section displays the following MTA-specific fields:

Field	Description
Serial Number	Displays the manufacturer's serial number for this MTA.
Hardware Version	Displays the manufacturer's hardware version for this MTA.
MAC Address	Displays the telephony MAC address for this MTA
Fully Qual Domain Name	Displays the fully qualified domain name (FQDN) for this MTA.
EndPoint Count	Displays the number of physical end points for this MTA.
Voice Enabled	Displays the MTA administrative status for this device. The options are: ■ True – Voice is enabled. ■ False – Voice is disabled.
Type ID	Displays the device type identifier for the DHCP option 60 exchanged between the MA and the DHCP server.
Prov State	Indicates the completion state of the provisioning process: ■ Pass – Pass state occurs after completing the processing of the configuration file. ■ In Progress – Occurs from boot time until configuration file processing is complete. ■ Fail – Pass state occurs after completing the processing of the configuration file. Manual intervention is required.
HTTP Access	Indicates whether HTTP file access is supported for MTA configuration file transfer.

Related Commands

- **show mta <mac-address> ncs** on page 481
- **show mta <mac-address> security** on page 485
- **show mta <mac-address> server** on page 487

show mta <mac-address> ncs

Displays Network Call Signaling (NCS) parameters for the MTA.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
show mta <mac-address> ncs {config | codec | endpoint}
[read-community <string>]
```

Arguments

<mac-address>	Specifies the MAC address of the MTA in xx:xx:xx:xx:xx:xx format.
{**config	codec	endpoint**}	Specifies one of the following values: ■ config – Displays NCS service-level configuration parameters. ■ codec – Displays types of CODECs that the MTA supports. ■ endpoint – Displays endpoints associated with the MTA.
read-community <string>	Specifies an optional community string for read access to the NCS information.

Example

```
cli:interface:cable:csi(1/1/1)# show mta 00:20:40:c0:4a:ba ncs config
S l o t                                  1
MAC Address                 00:20:40:c0:4a:ba
IP Address                        172.16.1.33
SID                                     115
CID                                       0
CPE                                       0
D:U                                     1:1
Power                                     0  (dbMV)
Timing                                ******
Modem State                       Registered

Echo Cancellation                       True
Silence Suppression                     True
Connection Mode             voice,fax,modem
R0 Cadence                  Intervals:
R6 Cadence                  Intervals: 1,2,4,5,6
R7 Cadence                  Intervals: 1,2,3,4,5,6
Def Call Signal TOS                       4
Def Media Stream TOS                      5
TOS Format Selector              Ipv4TOSOctet

cli:interface:cable:csi(1/1/1)# show mta 00:20:40:c0:4a:ba ncs codec
S l o t                                  1
MAC Address                 00:20:40:c0:4a:ba
IP Address                        172.16.1.33
SID                                     115
CID                                       0
CPE                                       0
D:U                                     1:1
Power                                     0  (dbMV)
Timing                                ******
Modem State                       Registered


row count: 10

Index    Type
-------- ----------
       1       g729
       2      g729a
       3      g729e
       4     g711mu
       5       g726
       6       g728
```

```
cli:interface:cable:csi(1/1/1)# show mta 00:20:40:c0:4a:ba ncs endpoint
S l o t                                    1
MAC Address                 00:20:40:c0:4a:ba
IP Address                        172.16.1.33
SID                                       115
CID                                         0
CPE                                         0
D:U                                       1:1
Power                                       0  (dbMV)
Timing                                 ******
Modem State                         Registered


row count: 2

Call Agent ID    Call Agent UDP
                 Port
---------------- ----------------
                              2427
                              2427
```

The command output is divided into two sections. The top section displays information that you can display using the **show modem** command. The bottom section displays the following MTA-specific fields when you use the **config** argument:

Field	Description
Echo Cancellation	Displays whether echoes are cancelled (True or False). True indicates that echo cancellation is in use. False indicates that echo cancellation is not in use.
Silence Suppression	Displays whether silence is suppressed in the send direction (True or False). True indicates that silence suppression is enabled. False indicates that silence suppression is disabled.
Connection Mode	Displays the various ways in which the MTA can connect to the network (such as voice, fax, and modem).
R0 Cadence	Displays ring cadence intervals, where each bit represents a duration of 200 milliseconds (6 second total).
R6 Cadence	Displays ring cadence intervals, where each bit represents a duration of 200 milliseconds (6 second total).
R7 Cadence	Displays ring cadence intervals, where each bit represents a duration of 200 milliseconds (6 second total).
Def Call Signal TOS	Displays the default Type of Service (ToS) value for call signalling (signals for setting up calls) in the IP header.

Field	Description
Def Media Stream TOS	Displays the default Type of Service (ToS) value for media stream packets in the IP header. Audio and video streams are examples of media streams.
TOS Format Selector	Displays one of the following formats for the default call signalling and media stream ToS values: ■ dscpCodepoint – Specifies that the ToS field is treated as a Differentiated Service Code Point (DSCP). The ToS field in the IP header identifies the differentiated service per hop behavior, which enables intermediate routers to select packets and apply specific forwarding rules based on the value of the ToS byte. ■ ipv4TOSOctet – Specifies that the ToS field is treated as an IPv4 ToS octet. Networks can provide a specific level of service based on the octet value in the packet.

The bottom section displays the following types of CODECs supported when you use the **codec** argument:

Field	Description
Index	Displays the internal codec index value.
Type	Displays the types of codecs that the MTA can support (other, unknown, g729, g729a, g711mu, g726, and g728).

The bottom section displays the following fields of information for endpoints when you use the **endpoint** argument:

Field	Description
Call Agent ID	Displays the call agent name, which can be an FQDN or an IP address.
Call Agent UDP Port	Displays the call agent UDP port for call signalling.

Related Commands

- **show mta <mac-address> base** on page 479
- **show mta <mac-address> security** on page 485
- **show mta <mac-address> server** on page 487

show mta <mac-address> security

Displays the MTA's public key certificates and other security-related information.

Mode

`interface:cable:csi(<c/s/i>)`

Syntax

`show mta` <mac-address> **`security`** [**`read-community`** <string>]

Arguments

<mac-address>	Specifies the MAC address of the MTA in xx:xx:xx:xx:xx:xx format.
read-community <string>	Specifies an optional community string for read access to the security information.

Example

```
cli:interface:cable:csi(1/4/1)# show mta 00:20:40:c0:83:00 security
S l o t                                4
MAC Address               00:20:40:c0:83:00
IP Address                       12.2.4.57
SID                                     52
CID                                      0
CPE                                      0
D:U                                      3
D:U                                    1:3
Power                                    0  (dbMV)
Timing                                1292
Modem State                     Registered

Manuf Certificate         xxxxxxxxxxxxxxx
Certificate               xxxxxxxxxxxxxxx
Signature                 xxxxxxxxxxxxxxx
Correlation ID                           0
```

The command output is divided into two sections. The top section displays information that you can display using the **show modem** command. The bottom section displays the following MTA-specific fields when you use the **security** argument:

Field	Description
Manuf Certificate	Displays the MTA manufacturer's X.509 public key certificate. This certificate is necessary to validate the MTA's device certificate.
Certificate	Displays the MTA's X.509 public key device certificate that the manufacurer installed in the MTA.
Signature	Displays the unique signature created by the MTA for each SNMP inform, SNMP trap, or SNMP get response exchanged prior to enabling the SNMPv3 ASN.1 encoded digital signature.
Correlation ID	Displays the random value generated by the MTA for use in authorizing registration.

Related Commands

- **show mta <mac-address> base** on page 479
- **show mta <mac-address> ncs** on page 481
- **show mta <mac-address> server** on page 487

show mta <mac-address> server

Displays the following parameters that the MTA uses to initialize when it boots up:

- DHCP, DNS, and SNMP servers
- Boot state
- URL of the configuration file that the MTA downloads from the TFTP server or HTTP server

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
show mta <mac-address> server [read-community <string>]
```

Arguments

<mac-address>	Specifies the MAC address of the MTA in xx:xx:xx:xx:xx:xx format.
read-community <string>	Specifies an optional community string for read access to the server, boot state, and configuration file URL.

Example

```
cli:interface:cable:csi(1/1/1)# show mta 00:20:40:c0:4a:ba server
S l o t                                 1
MAC Address                 00:20:40:c0:4a:ba
IP Address                        172.16.1.33
SID                                       115
CID                                         0
CPE                                         0
D:U                                       1:1
Power                                       0  (dbMV)
Timing                                 ******
Modem State                         Registered

Boot State
DHCP Server                          172.16.2.1
DNS 1 Server                         172.16.2.1
DNS 2 Server                            0.0.0.0
Config File                 TFTP://[172.16.2.1]/Moto4500_g1.cfg
SNMP Entity                 [172.16.2.1
```

Note that the Boot State field displays one of the following states:

Boot State	Description
operational	Indicates that the MTA has finished loading and processing configuration parameters, and the CMTS has completed the registration exchange.
disabled	Indicates that the MTA was administratively disabled, possibly because the configuration file specified that the MTA is refused network access.
waitingForDHCPOffer	Indicates that the DHCP discover message was transmitted and no offer message was received.
waitingForDHCPResponse	Indicates that the DHCP request message was transmitted and no response message was received.
waitingForConfig	Indicates that the request for the configuration server was made and no response was received.
refusedByCmts	Indicates that the registration request/response exchange with the CMTS failed.
other	Indicates that the boot state was other than those listed above.
unknown	Indicates that the boot state is unknown.

Related Commands

- **show mta <mac-address> base** on page 479
- **show mta <mac-address> ncs** on page 481
- **show mta <mac-address> security** on page 485

show ospf

Use this command to display the current Open Shortest Path First (OSPF) configuration. You can use this command without arguments to display OSPF global configuration, or use arguments to display the following OSPF information:

- Areas
- Border Routers
- LSA Database Contents
- Interfaces
- Virtual interfaces
- Neighbors
- Virtual neighbors

Mode

Any.

Syntax

```
show ospf {
[area [area-id]] |
[border-routers] |
[database {
   [asbr-summary] |
   [summary] |
   [external] |
   [network] |
   [nssa] |
   [router]
}] |
[interface] |
[interface virtual] |
[neighbor] |
[neighbor virtual]
}
```

Arguments

area <area-id>	Displays all areas when the area ID is not specified. Displays details on a specific area when the area ID is specified.
border-routers	Displays interfaces configured as OSPF border routers.
asbr-summary	Displays the contents of the OSPF ASBR Summary LSA database.
summary	Displays the contents of the OSPF Summary LSA database.
database	Shows all contents of the LSA database.
database summary	Shows a summary of the contents of the LSA database.
external	Displays the contents of the OSPF external LSA database.
network	Displays the contents of the OSPF network LSA database.
nssa	Displays OSPF Not So Stubby Areas (NSSAs).
router	Displays OSPF router links.
interface	Displays the OSPF interfaces defined on the system.
interface virtual	Displays the OSPF virtual interfaces defined on the system.
neighbor	Displays a list of OSPF neighbors. When you specify the neighbor argument followed by the virtual argument, you display a list of OSPF neighbors on virtual interfaces.
neighbor virtual	Displays the OSPF neighbors on virtual interfaces.

Example

The following example displays OSPF global parameters:

```
cli:192.168.220.230:router:ospf# show ospf
Admin Status                          Enabled
TOS Support                             False
Router Id                         201.1.1.1
ABR Status                              False
ASBR Status                             False

Report ospf-nbr-state                 Enabled
Report ospf-virt-nbr-state            Enabled
```

Related Commands

- **ospf area** on page 307
- **router ospf** on page 359

show out-counters

Displays outbound counter statistics for all interfaces on the system.

Mode

Any.

Syntax

show out-counters

Example

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# show out-counters

Interface     Type           Out octets  Out unicast Out multicast Out broadcast
------------- -------------- ----------- ----------- ------------- -------------
    1 / 1 / 1   docsCableMac   911657438       20275      20483432           514
    1 / 1 / 2    docsCableDS   911657966       20275      20483444           514
    1 / 3 / 1          sonet           0           0             0             0
    1 / 3 / 2      sonetPath           0           0             0             0
    1 / 3 / 3            ppp           0           0             0             0
    1 / 8 / 1       ethernet           0           0             0             0
   1 / 11 / 1       ethernet           0           0             0             0
   1 / 11 / 2       ethernet           0           0             0             0
   1 / 11 / 3       ethernet           0           0             0             0
   1 / 11 / 4       ethernet           0           0             0             0
   1 / 11 / 5       ethernet           0           0             0             0
   1 / 11 / 6       ethernet           0           0             0             0
   1 / 11 / 7       ethernet           0           0             0             0
   1 / 11 / 8       ethernet           0           0             0             0
  129 / 1 / 0     net bridge           0           0             0             0
  129 / 1 / 1     net bridge           0           0             0             0

row count: 16
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **show in-counters** on page 413

show pll-state

Displays the phase lock loop state of the current interface.

Mode

```
interface:<type>:csi(<c/s/i>)
```

Syntax

```
show pll-state
```

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show pll-state
PLL State                              normal
PLL Value                                   0

cli:192.168.208.3:interface:cable:csi(1/1/1)#
```

Related Commands

- **pll-state** on page 315

show privacy auth

Displays the BPI+ authorization lifetime configuration and statistics.

Mode

Any.

Syntax

show [**interface cable** <c/s/i>] **privacy auth** [<mac>] {**stats** | **error**}

Arguments

interface cable <c/s/i>	The interface for which you want to display BPI+ configuration. Note that this argument is required unless you are in interface:cable:csi(<c/s/i>) mode.
<mac>	The MAC address of the cable modem for which you want to display BPI+ configuration.
{**stats** | **error**}	Indicates that you want to display the errors for BPI+ configuration.

Example 1

This example displays the BPI+ authorization configuration for a specified interface.

```
cli:interface:cable:csi(1/1/1)# show privacy auth

row count: 2

MAC Address       SAID BPI     Expires Old          Cm     Cm     Cm Reset
                       Version Expires New          Life   Grace
                                                    Time   Time

----------------- ---- ------- -------------------- ------ ------ --------
00:90:83:36:82:ee   11 bpiplus 01-08-28 13:07:13    604800   600  noResetReque
                               01-09-04 13:07:13         0
```

Example 2

This example displays the BPI+ authorization lifetime configuration for a specified cable modem.

```
cli:interface:cable:csi(1/1/1)# show privacy auth 00:90:83:36:82:ee
MAC Address                00:90:83:36:82:ee
SAID                                       2
BPI Version                          bpiplus
Expires Old                01-08-28 13:07:13
Expires New                01-09-04 13:07:13
Cm Life Time                          604800
Cm Grace Time                            600
Cm Reset                  noResetRequested
```

Related Commands

- **privacy auth** on page 329

show privacy base

Displays the current BPI base configuration.

Mode

Any.

Syntax

show [**interface cable** <c/s/i>] **privacy base**

Arguments

interface cable <c/s/i>	Specifies the chassis/slot/interface for which you want to display the BPI base configuration. Note that this argument is required unless you are in interface:cable:csi(<c/s/i>) mode.

Example

This example shows the current BPI+ base configuration for a specified interface.

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show privacy base
Auth Life Time                          604800
TEK Life Time                            43200
Signed Manuf Cert Trust                trusted
Cert Validity Periods                    false
Authent Info                                 4
Auth Requests                                5
Auth Replies                                 4
Auth Rejects                                 0
Auth Invalids                                1
SA Map Requests                              0
SA Map Replies                               0
SA Map Rejects                               0

Encryption Mode                     56-bit-des
```

The fields in the command output are as follows:

Field	Description
Auth Life Time	Default lifetime, in seconds, the CMTS assigns to a new authorization key. The range is 1 to 604800.
TEK Life Time (Seconds)	Default lifetime, in seconds, the CMTS assigns to a new Traffic Encryption Key (TEK). The range is 1 to 604800.
Signed Manuf Cert Trust	Indicates whether signed manufacturers' certificates are trusted or untrusted by default.
Cert Validity Periods	Indicates whether certificate validity periods are enabled (true) or disabled (false).
Authent Info	Number of times the CMTS receives an authentication message from any cable modem.
Auth Requests	Number of times the CMTS receives an authorization request message from any cable modem.
Auth Replies	Number of times the CMTS transmits an authorization reply message to any cable modem.
Auth Rejects	Number of times the CMTS transmits an authorization reject message to any cable modem.
Auth Invalids	Number of times the CMTS transmits an authorization invalid message to any cable modem.
SA Map Requests	Number of times the CMTS receives an SA map request message from any cable modem.
SA Map Replies	Number of times the CMTS transmits an SA map reply message to any cable modem.
SA Map Rejects	Number of times the CMTS transmits an SA map reject message to any cable modem.
Encryption Mode	Encryption mode, such as 56-bit DES.

Related Commands

- **privacy base auth-lifetime** on page 331
- **privacy base cert-trust** on page 332
- **privacy base enable-cert-validity-periods** on page 333
- **privacy base tek-lifetime** on page 334

show privacy ca-cert

Displays manufacturer CA X.509 certificates. If you issue the command with no arguments, all certificates are displayed.

Mode

Any.

Syntax

show [**interface cable** <c/s/i>] **privacy ca-cert** [<number> [**details**]]

Arguments

interface cable <c/s/i>	Specifies the interface for which you want to display the certificates.
<number>	Specifies the index number of the X.509 certificate that you want to display. Values range from 1 to 10000.
details	Specifies that you want to display X.509 certificate details.

Example

```
cli:192.168.220.208:root# show privacy ca-cert

row count: 1

Index CA Cert   Thumbprint
      Trust
----- --------- --------------------
    1 untrusted

cli:192.168.220.208:root# show privacy ca-cert 1 details


Index:          1
Cert Trust:     untrusted
Thrumbprint:
CA Certificate:
[
[
  Version: V3
  Subject: CN=00:30:EB:FF:F0:44, OU="15110 Avenue of Science, San Diego, CA
9212
8", O=TurboNetComm, C=US
  Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5
  Key:  com.sun.rsajca.JSA_RSAPublicKey@503458
  Validity: [From: Fri Aug 18 03:00:00 EDT 2000,
              To: Tue Aug 18 03:00:00 EDT 2020]
  Issuer: CN=TurboNetComm Cable Modem Root Certificate Authority, OU="15110
Aven
ue of Science, San Diego, CA 92128", OU=DOCSIS, O=TurboNetComm, C=US
  SerialNumber: [    08000000 0000003c ]
]
  Algorithm: [SHA1withRSA]
  Signature:
0000: A3 38 37 11 9D 68 12 81   23 21 1D 5B E3 74 88 E7  .87..h..#!.[.t..
0010: F3 CB 24 1A 25 31 BE 12   4F C5 DB F8 57 2E 03 8C  ..$.%1..O...W...
0020: 4A 0B 5B E5 F2 14 62 7B   17 32 A6 4D E9 D5 82 65  J.[...b..2.M...e
0030: F2 46 06 3D 20 31 36 D1   03 F7 D1 F9 43 5C 34 E9  .F.= 16.....C\4.
0040: B9 D3 2E B1 DB 64 DB 54   FF 10 3E AF 72 4D A0 47  .....d.T..>.rM.G
0050: D2 BA 3B D7 70 CA 5F 85   32 86 B5 9C 62 BF EA F8  ..;.p._.2...b...
0060: 77 89 84 C9 60 09 D2 72   85 00 F3 73 FD 2E EA A1  w...`..r...s....
0070: 9C C0 A9 A3 86 30 53 00   6E 6C 69 13 7D 54 4C 00  .....0S.nli..TL.
]
-----BEGIN CERTIFICATE-----
MIICnTCCAgagAwIBAgIICAAAAAAADwwDQYJKoZIhvcNAQEFBQAwgaoxCzAJBgNVBAYT
AlVTMRUwEwYDVQQKEwxUdXJib05ldENvbW0xDzANBgNVBAsTBkRPQ1NJUzE1MDMGA1UE
```

```
CxMsMTUxMTAgQXZlbnVlIG9mIFNjaWVuY2UsIFNhbiBEaWVnbywgQ0EgOTIxMjgxPDA6
BgNVBAMTM1R1cmJvTmV0Q29tbSBDYWJsZSBNb2RlbSBSb290IENlcnRpZmljYXRlIEF1
dGhvcml0eTAeFw0wMDA4MTgwNzAwMDBaFw0yMDA4MTgwNzAwMDBaMHcxCzAJBgNVBAYT
AlVTMRUwEwYDVQQKEwxUdXJib05ldENvbW0xNTAzBgNVBAsTLDE1MTEwIEF2ZW51ZSBv
ZiBTY2llbmNlLCBTYW4gRGllZ28sIENBIDkyMTI4MRowGAYDVQQDExEwMDozMDpFQjpG
RjpGMDo0NDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAyhfsrMf8aeJJfYE8Bgse
4NR7N/hAP7Dbfc1DrzhlF36kEefNcmTSf+ii1Gp1fXS6BGnsnRV3PcRO5C/YVbKaQmGk
rn4cye6HfogpKqBNQPIgaUr5M/h6VmnxvWaeYO2I/2zZAkx1dkAALTLUwpQUoQuD94Zs
f6+qQt4NEAuXBdcCAwEAATANBgkqhkiG9w0BAQUFAAOBgQCjODcRnWgSgSMhHVvjdIjn
88skGiUxvhJPxdv4Vy4DjEoLW+XyFGJ7FzKmTenVgmXyRgY9IDE20QP30flDXDTpudMu
sdtk21T/ED6vck2gR9K6O9dwyl+FMoa1nGK/6vh3iYTJYAnScoUA83P9LuqhnMCpo4Yw
UwBubGkTfVRMAA==
-----END CERTIFICATE-----
```

Related Commands

- **privacy ca-cert** on page 335

show privacy cm-cert

Displays cable modem modem certificates and associated settings that are provisioned for the CMTS. If you issue the command with no arguments, all certificates are displayed.

Mode

Any.

Syntax

```
show [interface cable <c/s/i>] privacy cm-cert [<mac-address>]
{learnt | provisioned [details]}
```

Arguments

interface cable <c/s/i>	Specifies the interface for which you want to display the certificates. Note that this argument is required unless you are in interface:cable:csi(<c/s/i>) mode.
<mac-address>	Specifies the MAC address of the cable modem for which you want to display the certificates.
{**learnt** |	Indicates that you want to display the certificates in use by the cable modem.
provisioned	Indicates that you want to display the cable modem certificates that are provisioned for the CMTS.
[**details**]}	Displays the BPI+ privacy authorization details for the provisioned cable modem.

Example 1

```
mode: interface:cable:csi(1/1/1)
cli:192.168.208.3:interface:cable:csi(1/1/1)# show privacy cm-cert
Mac Address : 00:90:83:36:82:ee
Cert Invalid: validCmChained
Certificate :
[[
  Version: V3
  Subject: CN=00:90:83:36:82:EE, OU=1-1-1 Shibaura Minato-ku Tokyo,
O=Toshiba, C
=JP
  Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5
  Key:  com.sun.rsajca.JSA_RSAPublicKey@2e1f1f
  Validity: [From: Sat Apr 01 03:00:00 EST 2000,
              To: Wed Apr 01 03:00:00 EST 2020]
  Issuer: CN=Toshiba Cable Modem Root Certificate Authority, OU=1-1-1
Shibaura M
inato-ku Tokyo, OU=DOCSIS, O=Toshiba, C=JP
  SerialNumber: [    08000000 00008389 ]
]
  Algorithm: [SHA1withRSA]
  Signature:
0000: BF 7D F5 00 05 97 15 02   26 E0 03 CE D3 74 D2 9D  ........&....t..
0010: B9 C2 10 DE 2D F6 EC 4B   35 C9 90 A1 A9 6A 06 A6  ....-..K5....j..
0020: 72 51 74 85 0E 9C 46 64   38 0A 61 D5 2C 1F 74 D6  rQt...Fd8.a.,.t.
0030: 33 FD 27 1B D7 B9 ED 2B   3D 65 47 81 F4 1F DA 12  3.'....+=eG.....
0040: 8B C8 D6 48 B2 2B 50 7C   CF 84 5A B8 3A 11 AF F7  ...H.+P...Z.:...
0050: 4C 98 61 A5 52 2E BA F8   D2 79 85 3D 03 96 BE 7C  L.a.R....y.=....
0060: AB 30 1A 41 05 E8 D2 00   D7 CF 9F 00 F4 2B 99 32  .0.A.........+.2
0070: 02 F2 9C 84 55 90 D8 F4   C9 13 32 41 3C 22 72 B2  ....U.....2A<"r.
]
-----BEGIN CERTIFICATE-----
MIICcjCCAdugAwIBAgIICAAAAAAg4kwDQYJKoZIhvcNAQEFBQAwgZIxCzAJBgNVBAYT
AkpQMRAwDgYDVQQKEwdUb3NoaWJhMQ8wDQYDVQQLEwZET0NTSVMxJzAlBgNVBAsTHjEt
MS0xIFNoaWJhdXJhIE1pbmF0by1rdSBUb2t5bzE3MDUGA1UEAxMuVG9zaGliYSBDYWJs
ZSBNb2RlbSBSb290IENlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0wMDA0MDEwODAwMDBa
Fw0yMDA0MDEwODAwMDBaMGQxCzAJBgNVBAYTAkpQMRAwDgYDVQQKEwdUb3NoaWJhMScw
JQYDVQQLEx4xLTEtMSBTaGliYXVyYSBNaW5hdG8ta3UgVG9reW8xGjAYBgNVBAMTETAw
OjkwOjgzOjM2OjgyOkVFMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQD4gYpmziwm
h+Zuz15NKLZc/TNkYG8/XvBXCSV8hG4VAQLbTBNnC0MVzInMF/RZ9qCuVqByWPKXbHeG
C5gEQwrzCCMyYepUg57nESx51PHeCgVHgj5y10dSSgxV8BqLgKAT+v4arvMDrzDuQ5fZ
MMMx7zUNbuq13ItDjDR4gpNSTQIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAL999QAFlxUC
JuADztN00p25whDeLfbsSzXJkKGpagamclF0hQ6cRmQ4CmHVLB901jP9JxvXue0rPWVH
gfQf2hKLyNZIsitQfM+EWrg6Ea/3TJhhpVIuuvjSeYU9A5a+fKswGkEF6NIA18+fAPQr
mTIC8pyEVZDY9MkTMkE8InKy
-----END CERTIFICATE-----
```

Example 2

```
cli:interface:cable:csi(1/1/1)# show privacy cm-cert provisioned details
Mac Address:    11:11:11:11:11:11
Cert Trust:     trusted
CM Certificate:
[
[
Version: V3
Subject: CN=00:30:EB:FF:F0:44,OU="15110 Avenue of Science, San Diego, CA 9212
8", O=TurboNetComm, C=US
  Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5
  Key:  com.sun.rsajca.JSA_RSAPublicKey@242792
  Validity: [From: Fri Aug 18 03:00:00 EDT 2000,
              To: Tue Aug 18 03:00:00 EDT 2020]
  Issuer: CN=TurboNetComm Cable Modem Root Certificate Authority, OU="15110
Aven
ue of Science, San Diego, CA 92128", OU=DOCSIS, O=TurboNetComm, C=US
  SerialNumber: [    08000000 0000003c ]
]
  Algorithm: [SHA1withRSA]
  Signature:
0000: A3 38 37 11 9D 68 12 81   23 21 1D 5B E3 74 88 E7  .87..h..#!.[.t..
0010: F3 CB 24 1A 25 31 BE 12   4F C5 DB F8 57 2E 03 8C  ..$.%1..O...W...
0020: 4A 0B 5B E5 F2 14 62 7B   17 32 A6 4D E9 D5 82 65  J.[...b..2.M...e
0030: F2 46 06 3D 20 31 36 D1   03 F7 D1 F9 43 5C 34 E9  .F.= 16.....C\4.
0040: B9 D3 2E B1 DB 64 DB 54   FF 10 3E AF 72 4D A0 47  .....d.T..>.rM.G
0050: D2 BA 3B D7 70 CA 5F 85   32 86 B5 9C 62 BF EA F8  ..;.p._.2...b...
0060: 77 89 84 C9 60 09 D2 72   85 00 F3 73 FD 2E EA A1  w...`..r...s....
0070: 9C C0 A9 A3 86 30 53 00   6E 6C 69 13 7D 54 4C 00  .....0S.nli..TL.
]
-----BEGIN CERTIFICATE-----
MIICnTCCAgagAwIBAgIICAAAAAAAADwwDQYJKoZIhvcNAQEFBQAwgaoxCzAJBgNVBAYT
AlVTMRUwEwYDVQQKEwxUdXJib05ldENvbW0xDzANBgNVBAsTBkRPQ1NJUzE1MDMGA1UE
CxMsMTUxMTAgQXZlbnVlIG9mIFNjaWVuY2UsIFNhbiBEaWVnbywgQ0EgOTIxMjgxPDA6
BgNVBAMTM1R1cmJvTmV0Q29tbSBDYWJsZSBNb2RlbSBSb290IENlcnRpZmljYXRlIEF1
dGhvcml0eTAeFw0wMDA4MTgwNzAwMDBaFw0yMDA4MTgwNzAwMDBaMHcxCzAJBgNVBAYT
AlVTMRUwEwYDVQQKEwxUdXJib05ldENvbW0xNTAzBgNVBAsTLDE1MTEwIEF2ZW51ZSBv
ZiBTY2llbmNlLCBTYW4gRGllZ28sIENBIDkyMTI4MRowGAYDVQQDExEwMDozMDpFQjpG
RjpGMDo0NDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAyhfsrMf8aeJJfYE8Bgse
4NR7N/hAP7Dbfc1DrzhlF36kEefNcmTSf+ii1Gp1fXS6BGnsnRV3PcRO5C/YVbKaQmGk
rn4cye6HfogpKqBNQPIgaUr5M/h6VmnxvWaeYO2I/2zZAkx1dkAALTLUwpQUoQuD94Zs
f6+qQt4NEAuXBdcCAwEAATANBgkqhkiG9w0BAQUFAAOBgQCjODcRnWgSgSMhHVvjdIjn
88skGiUxvhJPxdv4Vy4DjEoLW+XyFGJ7FzKmTenVgmXyRgY9IDE20QP30flDXDTpudMu
sdtk21T/ED6vck2gR9K6O9dwyl+FMoa1nGK/6vh3iYTJYAnScoUA83P9LuqhnMCpo4Yw
UwBubGkTfVRMAA==
-----END CERTIFICATE-----
```

Related Commands

- **privacy cm-cert** on page 337
- **show privacy auth** on page 493

show privacy multicast

Displays the following information:

- Mappings of multicast IP addresses and SAs
- Multicast SA for each cable modem

Mode

Any.

Syntax

```
show [interface cable <c/s/i>] privacy multicast
{ip [<index>] | mac [<mac-address>]}
```

Arguments

interface cable <c/s/i>	Specifies the cable interface for which you want to display the IP multicast SA mappings. Note that this argument is not required if you are in interface:cable:csi(<c/s/i>) mode.
ip [<index>]	Specifies that you want to display the IP multicast SA map table. Optionally, you can specify an index value to display a specific entry in the table. Index values range from 1 to 10000.
mac [<mac-address>]	Specifies that you want to display cable modems and their associated multicast SAs. Optionally, you can specify a cable modem MAC address to display details on a specific cable modem.

Example 1

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show privacy multicast ip

row count: 1

Index IP Address      IP Mask         SAID  SA Type Encrypt Alg  Authent
                                                                 Alg
----- --------------- --------------- ----- ------- ------------ -------
    1 239.2.2.2       255.255.255.255  8192 dynamic des56cbcMode    none

cli:192.168.208.3:interface:cable:csi(1/1/1)# show privacy multicast ip 1
Index                                    1
Address                     239.2.2.2
Mask                        255.255.255.255
SAID                                  8192
SA Type                            dynamic
Encrypt Alg                   des56cbcMode
Authent Alg                           none
SA Map Requests                          0
SA Map Replies                           0
SA Map Rejects                           0
Reject Error Code                        0
```

Example 2

```
cli:192.168.220.208:interface:cable:csi(1/1/1)# show privacy multicast mac

row count: 1

Multicast Cm Mac Address
Auth SAID
--------- -----------------
     8888 00:95:10:4b:6a:f1
```

Related Commands

- **privacy multicast ip** on page 339
- **privacy multicast mac** on page 341
- **show privacy multicast** on page 504

show privacy tek

Displays the lifetime assigned by the CMTS to a Traffic Encryption Key (TEK) and an associated SAID.

Mode

Any.

Syntax

show [**interface cable <c/s/i>**] **privacy tek** [<said>] [**stats**]

Arguments

interface cable <c/s/i>	Specifies the cable modem interface for which you want to display the TEK lifetime that is assigned by the CMTS. Note that this argument is not required if you are in interface:cable:csi(<c/s/i>) mode.
said	The DOCSIS 1.1 Baseline Privacy security identifier between the CMTS and the cable modem. The range of identifier values is 0 to 4294967295.
stats	Displays the statistics of the SAID associated to this TEK.

Example 1

This example displays the lifetime assigned to a TEK for a specified cable interface.

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show privacy tek

row count: 1

SAID   SA Type Encryption   Authention   Life   Tek    Sequence
               Algorithm    Algorithm    Time   Reset  Number
------ ------- ------------ ------------ ------ ------ --------
     2 none            none         none  43200  false        2
```

Example 2

This example displays the lifetime assigned to a TEK and its associated SAID for a specified cable interface.

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show privacy tek 2
SAID                                    2
SA Type                              none
Encryption Algorithm                 none
Authention Algorithm                 none
Life Time                           43200
Tek Reset                           false
Sequence Number                         2
```

Example 3

This example displays the statistics for a specific SAID.

```
cli:192.168.208.3:root# show interface cable 1/1/1 privacy tek 1 stats
SAID                                    1
Requests                             5605
Replies                              5605
Rejects                                 0
Invalids                                0
```

Related Commands

- **privacy tek** on page 342

show proxy-arp

Displays the status of proxy ARP on the current cable interface.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
show proxy-arp
```

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show proxy-arp
Simple Proxy Arp:                    Disable

cli:192.168.208.3:interface:cable:csi(1/1/1)#
```

Related Commands

- **no proxy-arp** on page 292
- **proxy-arp** on page 343

show qos

Use this command to verify QoS configuration for the current cable interface.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
show qos
```

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show qos
FIA = 1 / 1 / 0 / 2

row count: 2

QOS     Service  Max        Guaranteed Max        Max        Baseline Status
Profile Priority Upstream   Bandwidth  Downstream Upstream   Privacy
Index            Bandwidth             Bandwidth  TxBurst
------- -------- ---------- ---------- ---------- ---------- -------- -------
   1025        7          0          0          0          0        2        1
   1026        1      64000          0    1000000          0        2        1
```

Related Commands

- **show interface cable <c/s/i> qos** on page 428

show radius-server

Displays details on the RADIUS authentication server.

Mode

root

Syntax

show radius-server

Example

```
cli:192.168.208.3:root# show radius-server
radius-server host 201.1.1.1
cli:192.168.208.3:root#
```

Related Commands

- **radius-server** on page 346

show relay-to-syslog

Displays whether trace log events for a module are relayed to a syslog server.

Mode

slot

Syntax

show relay-to-syslog

Example

```
cli:192.168.208.3:slot(1/1)# show relay-to-syslog
TraceLogToSyslog                    Disable
```

show rip neighbors

Displays a list of RIP neighbors on all interfaces.

Mode

Any.

Syntax

show rip neighbors

Example

```
cli:192.168.208.3:root# show rip neighbors
row count: 1
RIP Interface   Neighbor IP     Neighbor   Last Update Status
IP Address      Address         Type       Received
--------------- --------------- ---------- -------- -----------
20.1.1.1        20.1.1.2        configured        0     Active
```

The following fields appear in each neighbor entry:

Field	Description
RIP Interface IP Address	IP address of the Cuda 12000's IP interface.
Neighbor IP Address	IP address of the neighbor.
Neighbor Type	The neighbor can be one of the following types: ■ configured – The neighbor was configured with the **ip rip neighbor** command or with CudaView. ■ discovered – The Cuda 12000 learned of the neighbor automatically through network updates.
Last Update Received	Number of seconds since the last RIP update was received from the neighbor.
Status	Current status of the neighbor.

Related Commands

- **ip rip neighbor** on page 229

show rip stats all

Displays statistics on RIP activity since the time of the last Cuda 12000 reboot.

Mode

Any

Syntax

show rip stats all

Example

```
cli:192.168.208.3:root# show rip stats all
Since Reset At             01-06-21 10:08:45
Enabled Interfaces                         1
Number of Neighbors                        1
Packets Received                           0
Packets Sent                            5542
Requests Received                          0
Requests Sent                              0
Responses Received                         9
Responses Sent                          5533
Routes Timed Out                           0
Short Pkts. Received                       0
Bad Versions Received                      0
MBZ Errors                                 0
Source Port Errors                         0
Invalid IP Addresses                       0
Received from Self                         0
```

The Since Reset At field displays the time of the last Cuda 12000 reboot. All statistics counters are from the time of the last Cuda 12000 reboot. The statistics are as follows:

Statistic	Description
Enabled Interfaces	Total number of IP interfaces that run RIP.
Number of Neighbors	Total number of RIP neighbors on all IP interfaces that run RIP.
Packets Received	Total number of RIP packets received on all interfaces.
Packets Sent	Total number of RIP packets sent on all interfaces.
Requests Received	Total number of RIP requests received on all interfaces. Routers send requests to other routers to ask for all or part of their routing tables. Routers that receive requests respond with response packets.
Requests Sent	Total number of RIP requests sent on all interfaces.
Responses Received	Total number of RIP responses received on all interfaces. Responses contain all or part of the contents of a router's routing table.
Responses Sent	Total number of RIP responses sent on all interfaces. The Cuda 12000 can send a response to a request or it can send a response unsolicited.
Routes Timed Out	Total number of RIP routes that have timed out. Each route in the routing table has a timeout timer, which is initialized when the Cuda 12000 first learns about the route. When that timer expires, the route is no longer valid. Timer expiration occurs when a route is not updated.
Short Pkts. Received	Total number of RIP packets received on all interfaces containing a RIP header size less than the legal RIP header size.
Bad Versions Received	Total number of RIP packets received on all interfaces containing an illegal version (that is, a version other than RIP Version 1 or RIP Version 2).
MBZ Errors	Total number of RIP packets received on all interfaces containing fields that must have a value of zero and are not set to zero. Note that these fields are referred to as must be zero (MBZ) fields.
Source Port Errors	Total number of RIP packets received on all interfaces which did not originate from UDP port 520.
Invalid IP Addresses	Total number of RIP packets received on all interfaces containing an invalid ip address.
Received from Self	Total number of RIP packets that the Cuda 12000 tries to send to itself on all interfaces.

Related Commands

- **show rip stats current** on page 515

show rip stats current

Displays statistics on RIP activity since the last time counters were reset with the **reset rip stats** command or the Cuda 12000 rebooted (whichever came last).

Mode

Any

Syntax

show rip stats current

Example

```
cli:192.168.208.3:root# reset rip stats
cli:192.168.208.3:root# show rip stats current
Since Reset At             01-06-21 10:08:46
Enabled Interfaces                         1
Number of Neighbors                        1
Packets Received                           0
Packets Sent                               0
Requests Received                          0
Requests Sent                              0
Responses Received                         0
Responses Sent                             0
Routes Timed Out                           0
Short Pkts. Received                       0
Bad Versions Received                      0
MBZ Errors                                 0
Source Port Errors                         0
Invalid IP Addresses                       0
Received from Self                         0
```

The Since Reset At field displays the time at which counters were reset with the **reset rip stats** command. All statistics counters are from the last time that the **reset rip stats** command was issued or the Cuda 12000 rebooted, whichever came last. The statistics are as follows:

Statistic	Description
Enabled Interfaces	Total number of IP interfaces that run RIP.
Number of Neighbors	Total number of RIP neighbors on all IP interfaces that run RIP.

Statistic	Description
Packets Received	Total number of RIP packets received on all interfaces.
Packets Sent	Total number of RIP packets sent on all interfaces.
Requests Received	Total number of RIP requests received on all interfaces. Routers send requests to other routers to ask for all or part of their routing tables. Routers that receive requests respond with response packets.
Requests Sent	Total number of RIP requests sent on all interfaces.
Responses Received	Total number of RIP responses received on all interfaces. Responses contain all or part of the contents of a router's routing table.
Responses Sent	Total number of RIP responses sent on all interfaces. The Cuda 12000 can send a response to a request or it can send a response unsolicited.
Routes Timed Out	Total number of RIP routes that have timed out. Each route in the routing table has a timeout timer, which is initialized when the Cuda 12000 first learns about the route. When that timer expires, the route is no longer valid. Timer expiration occurs when a route is not updated.
Short Pkts. Received	Total number of RIP packets received on all interfaces containing a RIP header size less than the legal RIP header size.
Bad Versions Received	Total number of RIP packets received on all interfaces containing an illegal version (that is, a version other than RIP Version 1 or RIP Version 2).
MBZ Errors	Total number of RIP packets received on all interfaces containing fields that must have a value of zero and are not set to zero. Note that these fields are referred to as must be zero (MBZ) fields.
Source Port Errors	Total number of RIP packets received on all interfaces which did not originate from UDP port 520.
Invalid IP Addresses	Total number of RIP packets received on all interfaces containing an invalid ip address.
Received from Self	Total number of RIP packets that the Cuda 12000 tries to send to itself on all interfaces.

Related Commands

- **reset rip stats** on page 351
- **show rip stats all** on page 513

show route-map

Displays a list of configured route maps.

Mode

To display import OSPF route maps:

```
router:ospf:import
```

To display export OSPF route maps:

```
router:ospf:export
```

To display import RIP route maps:

```
router:rip:import
```

To display export RIP route-maps:

```
router:rip:export
```

Syntax

show route-map [<map-id>]

Arguments

<map-id>	Specifies a route map ID.

Example

```
cli:192.168.208.3:root# router ospf import
mode: router:ospf:import
cli:192.168.208.3:router:ospf:import# show route-map

row count: 2

ID  Description Route Address   Route Mask      Peer Address    Peer Mask
--- ----------- --------------- --------------- --------------- -------------
  1 tryospfflte 0.0.0.0         0.0.0.0         0.0.0.0         0.0.0.0
    rs
  2 tryaddospf  0.0.0.0         0.0.0.0         0.0.0.0         0.0.0.0

cli:192.168.208.3:router:ospf:import# show route-map 1
ID                                    1
Description                tryospfflters
Route Address                    0.0.0.0
Route Mask                       0.0.0.0
Peer Address                     0.0.0.0
Peer Mask                        0.0.0.0
Tag                                   0
Key Bits                              0
Preference                            0
Flags                                 1
```

Related Commands

- **route-map** on page 354

show running-config

Displays the current system configuration on the Cuda 12000 in CLI command format or XML format. By default, the command displays the current system configuration in CLI command format.

When issued with no arguments, the command displays only configuration parameters that have a value other than the default. To display all configuration parameters, issue the command with the **all** argument.

Mode

Any.

Syntax

```
show running-config [{all | xml | server-name <server-name>}]
```

Arguments

{**all** | **xml** | **server-name** <server-name>}	Specifies one of the following values: ■ all – Displays all configuration parameters. The command output displays both parameters that have default values and parameters that have non-default values. ■ xml – Displays the current system configuration in XML format. ■ server-name <server-name> – For ADC internal use only.

Example 1

```
cli:192.168.220.208:root# show running-config | include snmp
  snmp-server group adc v1 read public write private notify public context adc
storage nonvolatile
  snmp-server group adc v2c read public write private notify public context
adc storage nonvolatile
  snmp-server group adc v3 noauth read public write private notify public
context adc storage nonvolatile
  snmp-server group guitraps v1 notify guitraps storage readonly
  snmp-server group guitraps v2c notify guitraps storage readonly
  snmp-server view public 1.3.6.1 included storage nonvolatile
  snmp-server view private 1.3.6.1 included storage nonvolatile
  snmp-server view guitraps 1.3.6.1 included storage readonly
  snmp-server community guitraps guitraps storage readonly
  snmp-server community private adc context adc storage nonvolatile
  snmp-server community public adc context adc storage nonvolatile
  snmp-server host 127.0.0.1 guitraps udp-port 54321 storage readonly
```

Example 2

```
cli:192.168.220.208:root# show running-config all | include trace-log
  trace-log cfm critical
  trace-log ma critical
  trace-log la critical
  trace-log rm critical
  trace-log ldp critical
  trace-log cmts critical
  trace-log cmts-mac critical
  trace-log cmts-bpi critical
  trace-log snmp critical
  trace-log agentx critical
  trace-log dhcp-relay critical
  trace-log mal critical
  trace-log java-server critical
  trace-log cfg-rmi critical
  trace-log prov-rmi critical
  trace-log faults-rmi critical
  trace-log ldap-client critical
  trace-log jni critical
  trace-log ca critical
  trace-log rbp critical
  trace-log crp critical
  trace-log ftd critical
  trace-log rip critical
--More--
```

show shared-secret

Displays the shared secret for the current interface.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
show shared-secret
```

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show shared-secret
Shared Key
     ASCII: "¦4V"
     HEX:   "12:34:56"
Shared Key Enabled                      Yes
```

Related Commands

- **shared-secret** on page 367

show snmp

Displays SNMP summary details.

Mode

root

Syntax

show snmp

Example

```
cli:192.168.244.212:root# show snmp
Contact                    No Contact
Name                       No Name
Location                   No Location
SNMP packets received                 236016
Bad SNMP version errors                    0
Unknown community names                    0
Illegal community names                    0
Encoding errors                            0
Silent drops                               0
Unknown security models                    0
Invalid messages                           0
Unknown PDU handlers                       0
Authentication traps                 disable
```

Related Commands

- **show snmp community** on page 523
- **show snmp context** on page 525
- **show snmp group** on page 527
- **show snmp host** on page 528
- **show snmp notify** on page 530
- **show snmp user** on page 532
- **show snmp view** on page 533

show snmp community

Displays SNMP communities.

Mode

`root`

Syntax

`show snmp community` [<community-name>]

Arguments

<community-name> Name that identifies the SNMP community that you want to display.

Example 1

This example is a display of all the communities:

```
cli:192.168.244.212:root# show snmp community

row count: 3

Name                    Security Name           Context             Storage
----------------------- ----------------------- ------------------- ---------
guitraps                guitraps                                    NonVolatile
private                 adc                     adc                 NonVolatile
public                  adc                     adc                 NonVolatile
```

Example 2

This example is a display of a specific community:

```
cli:192.168.208.3:root# show snmp community private

row count: 1

Name         Security Name    Context      Storage
------------------------ ------------------------------
private      adc              adc          NonVolatile
```

Related Commands

- **show snmp context** on page 525
- **show snmp group** on page 527
- **show snmp host** on page 528
- **show snmp notify** on page 530
- **show snmp user** on page 532
- **show snmp view** on page 533
- **snmp-server community** on page 556

show snmp context

Displays the SNMP contexts for accessing management information.

Mode

```
root
```

Syntax

```
show snmp context [<context-name>]
```

Arguments

<context-name>	Name of the context that was assigned by the **snmp-server context** command.

Example

```
cli:192.168.208.3:root# show snmp context adc

row count: 1

Name       Storage      Status
-------------------------------
adc        NonVolatile  Active
```

Related Commands

- **no snmp-server context** on page 294
- **show snmp community** on page 523
- **show snmp group** on page 527
- **show snmp host** on page 528
- **show snmp notify** on page 530
- **show snmp user** on page 532
- **show snmp view** on page 533
- **snmp-server context** on page 559

show snmp engineID

Displays the SNMP engine's unique identifier and the number of times the SNMP engine initialized since the last SNMP Engine ID configuration. The SNMP engine's unique identifier is a 14-byte octet string. Note that the SNMP engine replaces the SNMP agent and SNMP manager from earlier versions of SNMP.

Mode

```
root
```

Syntax

```
show snmp engineID
```

Example

```
cli:192.168.220.206:root# show snmp engineID
Local SNMP engineID        00000da5005057000519a081bca0
SNMP engine boots                          29
```

Related Commands

- **show snmp community** on page 523
- **show snmp context** on page 525
- **show snmp group** on page 527
- **show snmp host** on page 528
- **show snmp notify** on page 530
- **show snmp user** on page 532
- **show snmp view** on page 533

show snmp group

Displays SNMP security groups.

Mode

`root`

Syntax

`show snmp group` [<group-name>]

Arguments

<group-name> Name of the group assigned by the **snmp-server group** command.

Example

```
cli:192.168.208.3:root# show snmp group team

row count: 1

Group      Context Model Level  Read View   Write View  Notify View Storage
---------- ------- ----- ------ ----------- ----------- ----------- ---------
team               V3    Auth   v1default                          NonVolatile

cli:192.168.208.3:root#
```

Related Commands

- **show snmp community** on page 523
- **show snmp context** on page 525
- **show snmp host** on page 528
- **show snmp notify** on page 530
- **show snmp user** on page 532
- **show snmp view** on page 533
- **snmp-server group** on page 561

show snmp host

Displays SNMP hosts on the trap recipient list and their SNMP community membership. When issued with no arguments, the command displays all SNMP hosts defined with the **snmp-server host** command.

Mode

`root`

Syntax

`show snmp host` [<ip-address>] [**`parameters`**]

Arguments

<ip-address>	IP address of an SNMP host.
parameters	Displays details on SNMP host parameters.

Example

```
cli:null:root> enable rootcli:192.168.208.3:root# show snmp host

row count: 2

Host:Port           Time Retry Notify or      Storage     Mask            MMS
                    -out       Communities
------------------- ---- ----- -------------- ----------- --------------- ---
201.100.1.1:164      500   200 inform:201.100 NonVolatile                5000
                               .1.1:164
127.0.0.1:54321       15     3 trap:127.0.0.1 NonVolatile                 484
                               :54321

cli:192.168.208.3:root# show snmp host 201.100.1.1

row count: 1

Host:Port           Time Retry Notify or      Storage     Mask            MMS
                    -out       Communities
------------------- ---- ----- -------------- ----------- --------------- ---
201.100.1.1:164      500   200 inform:201.100 NonVolatile                5000
                               .1.1:164

cli:192.168.208.3:root# show snmp host 201.100.1.1 parameters

row count: 1

Notify:Host:Port             Model Level  Group Name                Storage
---------------------------- ----- ------ ------------------------ ---------
inform:201.100.1.1:164       V2c   NoAuth private                  NonVolatile
```

Note that the 127.0.0.1:54321 host entry is a special entry that CudaView uses to receive traps from the chassis.

Related Commands

- **show snmp community** on page 523
- **show snmp context** on page 525
- **show snmp group** on page 527
- **show snmp notify** on page 530
- **show snmp user** on page 532
- **show snmp view** on page 533
- **snmp-server host** on page 564

show snmp notify

Displays SNMP hosts and their notification parameters. When issued with no arguments, the command displays all hosts and their notification parameters.

Mode

`root`

Syntax

`show snmp notify` [<ip-address>]

Arguments

<ip-address>	IP address of an SNMP host.

Example

```
cli:null:root> enable rootcli:192.168.208.3:root# show snmp notify

row count: 2

Host:Port            Storage     Notifications Sent                       Type
-------------------- ----------- ---------------------------------------- ----
201.100.1.1:164     NonVolatile cold-start link-up                      inform
127.0.0.1:54321      NonVolatile docs-init-req-fail docs-init-rsp-fail   V2
                                 docs-init-ack-fail docs-dyn-req-fail
                                 docs-dyn-rsp-fail docs-dyn-ack-fail
                                 docs-bpi-init docs-bpkm docs-dynamic-sa
                                 docs-dcc-req-fail docs-dcc-rsp-fail
                                 docs-dcc-ack-fail prov-service
                                 ldap-failed ldap-restored
                                 subnet-addr-low subnet-addr-high
                                 isp-addr-low isp-addr-high
                                 duplicate-addr bcm-failover-down
                                 bcm-failover-up bcm-sw-mismatch
                                 card-down card-up trace-log
                                 cable-modem-up cable-modem-down
                                 bcm-state-change icl-state-change
                                 cable-modem-auth-failure
                                 dhcp-relay-not-configured
                                 local-sonet-alarm remote-sonet-alarm
                                 chassis-fault chassis-fault-cleared
                                 cold-start warm-start link-down link-up
                                 authentication-failure
```

Note that the 127.0.0.1:54321 host entry is a special entry that CudaView uses to receive traps from the chassis.

Refer to Table 2-3 on page 566 for descriptions of notifications.

Related Commands

- **show snmp community** on page 523
- **show snmp context** on page 525
- **show snmp group** on page 527
- **show snmp host** on page 528
- **show snmp user** on page 532
- **show snmp view** on page 533

show snmp user

Displays SNMP user accounts. When issued with no arguments, the command displays all SNMP user accounts.

Mode

root

Syntax

show snmp user [<user>]

Arguments

<user>	Name of the user account as specified by the **snmp-server user** command.

Example

```
cli:192.168.220.207:root# show snmp user
row count: 2
Name                           Authentication Privacy Storage     Status
------------------------------ -------------- ------- ----------- ---------
adc                            HMAC-MD5-96    CBC-DES NonVolatile Active
template                       HMAC-MD5-96    CBC-DES Permanent   Active
cli:192.168.220.207:root# show snmp user adc
row count: 1
Name                           Authentication Privacy Storage     Status
------------------------------ -------------- ------- ----------- ----------
adc                            HMAC-MD5-96    CBC-DES NonVolatile Active
```

Related Commands

- **show snmp community** on page 523
- **show snmp context** on page 525
- **show snmp group** on page 527
- **show snmp host** on page 528
- **show snmp notify** on page 530
- **show snmp view** on page 533
- **snmp-server user** on page 574

show snmp view

Displays SNMP access views. When issued with no arguments, the command displays all views.

Mode

`root`

Syntax

`show snmp view` [<view-name>]

Arguments

<view-name>	Name of the SNMP view as assigned by the **snmp-server view** command.

Example

```
cli:192.168.220.206:root# show snmp view

row count: 3

View Name        Subtree                     Type     Storage     Status
---------------- --------------------------- -------- ----------- -----------
public           1.3.6.1                     Included NonVolatile Active
private          1.3.6.1                     Included NonVolatile Active
guitraps         1.3.6.1                     Included NonVolatile Active

cli:192.168.220.206:root# show snmp view guitraps

row count: 1

View Name        Subtree                     Type     Storage     Status
---------------- --------------------------- -------- ----------- -----------
guitraps         1.3.6.1                     Included NonVolatile Active

cli:192.168.220.206:root#
```

Related Commands

- **show snmp community** on page 523
- **show snmp context** on page 525
- **show snmp group** on page 527
- **show snmp host** on page 528
- **show snmp notify** on page 530
- **show snmp user** on page 532
- **snmp-server view** on page 576

show spectrum-group

The **show spectrum-group** commands allow you to display various information about the frequency hopping configuration on the upstream channels.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
show spectrum-group upstream
show spectrum-group <rule number> upstream
show spectrum-group upstream <channel number>
show spectrum-group <rule number> upstream <channel number>
show spectrum-group stats upstream <channel number>
show spectrum-group stats upstream
```

Arguments

<rule number>	The user-defined number assigned to a policy configuration (1, 2, 3, 4, or 5). Five policies may be configured for each upstream channel.
<channel number>	The upstream channel port on which the policy is set. Valid range is 1 to 4 for 1x4 modules and 1 to 6 for 1x6 modules.

Example 1

This example displays the policy configuration for all upstream channels.

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show spectrum-group upstream
row count: 4
Rule   Upstream Threshold Interval Frequency  Profile
Number ID                         (MHz)      Num
------ -------- --------- -------- ---------- -------
     1        1         1       10        5.0       1
     2        1         1       10        5.0       1
     3        1         1       10        5.0       1
     2        2         1    86400       42.0       1
cli:192.168.208.3:interface:cable:csi(1/1/1)#
```

Example 2

This example displays the Policy Number 2 configuration for all upstream channels.

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show spectrum-group 2 upstream
row count: 2
Rule   Upstream Threshold Interval Frequency  Profile
Number ID                         (MHz)      Num
------ -------- --------- -------- ---------- -------
     2        1         1       10        5.0       1
     2        2         1    86400       42.0       1
cli:192.168.208.3:interface:cable:csi(1/1/1)#
```

Example 3

This example displays a policy configuration for upstream channel number 1.

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show spectrum-group upstream 1
row count: 3
Rule   Threshold Interval Frequency  Profile
Number                    (MHz)      Num
------ --------- -------- ---------- -------
     1         1       10        5.0       1
     2         1       10        5.0       1
     3         1       10        5.0       1
Current Rule: 1
cli:192.168.208.3:interface:cable:csi(1/1/1)#
```

Example 4

This example displays a policy configuration statistics for upstream channel number 2.

```
cli:interface:cable:csi(1/1/1)# show spectrum-group stats upstream 2
row count: 1
Rule   Threshold Interval Frequency  Profile Error Error Count Error Packet
Number                    (MHz)      Num     Rate              Count
------ --------- -------- ---------- ------- ----- ----------- ------------
     2         1    86400       42.0       1     0           0            0
```

Example 5

This example displays policy configuration statistics for all upstream channels.

```
cli:interface:cable:csi(1/1/1)# show spectrum-group stats upstream

row count: 4

Rule   Upstream Threshold Interval Frequency  Profile Error Error Count Err Pk
Number ID                          (MHz)      Num     Rate              Count
------ -------- --------- -------- ---------- ------- ----- ----------- -----
     1        1         1       10        5.0       1     0           0 216
     2        1         1       10        5.0       1     0           0 0
     3        1         1       10        5.0       1     0           0 0
     2        2         1    86400       42.0       1     0           0 0
cli:interface:cable:csi(1/1/1)#
```

Related Commands

- **spectrum-group** on page 579
- **spectrum-group reset upstream** on page 581

show syslog-server

Use this command to display which facilities the syslog server is configured to receive messages from and at which debug level.

Mode

Any.

Syntax

show syslog-server

Example

```
cli:172.16.19.10:root# show syslog-server
Facility Level    Action                  Priority Action Value
-------- -------- ----------------------- -------- ------------------------
All      Debug    Log To File             175      /bas/log/messages

cli:172.16.19.10:root#
```

show system

Display system uptime for all active application modules.

Mode

To display system uptime for all system modules:

`root`

To display system uptime for a specific slot:

`slot(<c/s>)`

Syntax

`show system`

Example

```
cli:172.16.19.10:root# show system

row count: 5

Chassis Slot    Boot Time          Up Time (duration)
------- ------- -------------------------------------------------
      1       1 99-09-08 17:10:58  1 hrs 50 mins 44 secs
      1       3 99-09-08 17:06:09  1 hrs 55 mins 33 secs
      1       8 99-09-08 17:06:13  1 hrs 55 mins 29 secs
      1      11 99-09-08 17:07:35  1 hrs 54 mins 7 secs
      1      11 99-09-08 17:07:37  1 hrs 54 mins 5 secs

cli:172.16.19.10:root#
```

show tacacs-server

Displays the current IP address and encryption key that is configured on the TACACS+ server.

Mode

root

Syntax

show tacacs-server

Example

```
cli:192.168.208.3:root# show tacacs-server
tacacs-server host 192.168.208.3
tacacs-server key cuda12000
cli:192.168.208.3:root#
```

Related Commands

- **aaa authentication** on page 44
- **show aaa** on page 368
- **tacacs-server** on page 585

show time

Displays the system time.

Mode

Any.

Syntax

show time

Example

```
cli:192.168.208.3:root# show time
Host system time:   Wed Sep 05 13:18:02 EDT 2001
```

Related Commands

- **set time** on page 365

show timeout

Displays the timeout for idle CLI sessions.

Mode

Any.

Syntax

```
show timeout
```

Example

```
cli:192.168.208.3:root# show timeout
Time Out: 15 Minutes
```

Related Commands

- **set timeout** on page 366

show topology

Displays the current system topology. This display lists all modules that are installed in the system, the physical ports, and associated status.

Mode

Any.

Syntax

show topology

Example

```
cli:192.168.208.3:root# show topology

row count: 10

Chassis/Slot/    Class      Interface Type       Status
Interface
---------------- ---------- -------------------- --------------
1 / 1 / 1        Egress     docsCableMaclayer    Active
1 / 8 / 1        Egress     POS (OC3c)           Not In Service
1 / 11 / 1       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 2       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 3       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 4       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 5       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 6       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 7       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 8       Egress     Ethernet (100 Mb)    Not In Service

cli:192.168.208.3:root#
```

show trace-log

Shows the trace-log configuration for the current slot. Trace-log configuration determines the information displayed within basmonitor.

Mode

```
slot(<c/s>)
```

Syntax

```
show trace-log
```

Example

```
cli:172.16.19.10:slot(1/1)# show trace-log

row count: 78

SW Component    Log Level
--------------- ---------
            cfm  critical
             ma  critical
             la  critical
             rm  critical
            ldp  critical
           cmts      info
       cmts-mac  critical
       cmts-bpi  critical
           snmp  critical
         agentx  critical
     dhcp-relay      info
            mal  critical
-more-
```

Related Commands

- **basmonitor** on page 66
- **trace-log** on page 587

show traffic-relay

Displays the traffic relay status of servers and other processes on the Cuda 12000. Processes that have traffic relay enabled can send and receive traffic using an internal address. Otherwise, processes cannot perform internal traffic forwarding.

Issuing this command with no arguments displays status for all processes. Issuing this command with a specific argument displays details on the specified process only.

Mode

`root`

Syntax

```
show traffic-relay [{dns | ftp | http | snmp | snmp-trap | ssh |
  syslog | telnet | tftp | time_of_day}]
```

Arguments

dns	Displays details on DNS internal traffic forwarding.
ftp	Displays details on FTP internal traffic forwarding.
http	Displays details on HTTP internal traffic forwarding.
snmp	Displays details on SNMP management internal traffic forwarding.
snmp-trap	Displays details on SNMP trap internal traffic forwarding.
ssh	Displays details on secure shell internal traffic forwarding.
syslog	Displays details on syslog internal traffic forwarding.
telnet	Displays details on Telnet internal traffic forwarding.
tftp	Displays details on TFTP internal traffic forwarding.
time_of_day	Displays details on Time-of-Day internal traffic forwarding.

Example

```
cli:192.168.208.3:root# show traffic-relay

row count: 10

Protocol     State    Port Number
------------ -------- -----------
tftp         enable            69
time_of_day  enable            37
syslog       enable           514
dns          enable            53
snmp         disable          161
telnet       disable           23
ssh          disable           22
http         enable            80
ftp          disable           21
snmp-trap    enable           162

cli:192.168.208.3:root# show traffic-relay tftp
Protocol                                tftp
State                                 enable
Port Number                               69
```

Related Commands

- **traffic-relay** on page 596

show upstream

Displays upstream map parameters, upstream ranging parameters, upstream status, and other upstream information for the current interface.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
show upstream {map | ranging}
```

Arguments

{**map** | **ranging**}	Specifies one of the following values: ■ map – Displays upstream map information. ■ ranging – Displays upstream range information.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show upstream ranging
Upstream Channel Id        1 (1 / 1 / 3 / 2)
Power Offset Threshold                    8  (1/4 dB)
Power Desired                          8192
Max Ignored Invitations                  16
Init Ranging Timeout                    400  (millisecs)
Maximum Power Adjustment                 24  (1/4 dB)
Zero Power Adjustment                 False
Zero Timing Adjustment                False
Zero Frequency Adjustment             False

Upstream Channel Id        2 (1 / 1 / 4 / 2)
Power Offset Threshold                    8  (1/4 dB)
Power Desired                          4096
Max Ignored Invitations                  16
Init Ranging Timeout                    400  (millisecs)
Maximum Power Adjustment                 24  (1/4 dB)
Zero Power Adjustment                 False
Zero Timing Adjustment                False
Zero Frequency Adjustment             False

Upstream Channel Id        3 (1 / 1 / 5 / 2)
Power Offset Threshold                    8  (1/4 dB)
Power Desired                          4096
Max Ignored Invitations                  16
Init Ranging Timeout                    400  (millisecs)
Maximum Power Adjustment                 24  (1/4 dB)
Zero Power Adjustment                 False
Zero Timing Adjustment                False
Zero Frequency Adjustment             False

Upstream Channel Id        4 (1 / 1 / 6 / 2)
Power Offset Threshold                    8  (1/4 dB)
Power Desired                          4096
Max Ignored Invitations                  16
Init Ranging Timeout                    400  (millisecs)
Maximum Power Adjustment                 24  (1/4 dB)
Zero Power Adjustment                 False
Zero Timing Adjustment                False
Zero Frequency Adjustment             False

        Timer: elapsed time: 9394 mSeconds
```

Related Commands

- **upstream <port number> channel-width** on page 600
- **upstream <port number> data-backoff** on page 601
- **upstream <port number> frequency** on page 602
- **upstream <port number> map** on page 603
- **upstream <port number> minislot-size** on page 605
- **upstream <port number> modulation-profile** on page 607
- **upstream <port number> no shutdown** on page 608
- **upstream <port number> power-level** on page 609
- **upstream <port number> range-backoff** on page 610
- **upstream <port number> ranging** on page 611
- **upstream <port number> shutdown** on page 613
- **upstream <port number> voice-bw-reserve** on page 614

show user

Use this command to display the current account name.

Mode

root

Syntax

show users

Example

```
cli:172.16.19.10:root# show user
current user: root
cli:172.16.19.10:root# show active-user
Displaying active users ...
SessionId: CLI:32a8ff8e:da1f1a9ffd:-7f62     UserName: root
cli:172.16.19.10:root#
```

Related Commands

- **account** on page 621
- **show active-user** on page 626

show version

Use this command to determine the software revisions for all modules in the Cuda 12000 chassis.

Mode

Any.

Syntax

```
show version
```

Example

```
cli:192.168.208.3:root# show version
3.0.16 Release3.0_Beta 150 2001_08_28_1127

row count: 5

Chassis Slot    LPort   Boot Time      Description
------- ------- ------- -------------- ---------------------------------------
      1       1       2 01-08-29 18:25 BAS CMTS 1X4, Hardware V1 (serial
                                        #0000000495), Software V3.0, Build #16
                                        [Release3.0_Beta  150  ] built
                                        2001_08_28_1127 Pentium 399 MHz, Flash
                                       N/A, SDRAM 64MB;SA1200 B0:166MHz, FLASH
                                        2MB, SDRAM 128MB
      1       3       1 01-08-29 18:25 BAS Forwarder, Hardware V1 (serial
                                        #0000000673), Software V3.0, Build #16
                                        [Release3.0_Beta  150  ] built
                                       2001_08_28_1127 SA1200 B0:166MHz, Flash
                                        2MB, SDRAM 128MB
      1       8       1 01-09-03 13:58 BAS Forwarder, Hardware V1 (serial
                                        #0000001437), Software V3.0, Build #16
                                        [Release3.0_Beta  150  ] built
                                       2001_08_28_1127 SA1200 B0:166MHz, Flash
                                        2MB, SDRAM 128MB
      1      11       1 01-08-29 18:26 BAS Forwarder, Hardware V1 (serial
                                        #0000000000), Software V3.0, Build #16
                                        [Release3.0_Beta  150  ] built
                                       2001_08_28_1127 SA1200 B0:166MHz, Flash
                                        2MB, SDRAM 128MB
      1      11       2 01-08-29 18:25 BAS Route Server, Hardware V1 (serial
                                        #0000000000), Software V3.0, Build #16
                                        [Release3.0_Beta  150  ] built
                                        2001_08_28_1127 Pentium 400 MHz, Flash
                                        N/A, SDRAM 64MB
```

shutdown

Use this command to administratively disable an interface. Use the no form of this command (**no shutdown**) to enable the interface.

Mode

```
interface:<type>:csi(<c/s/i>)
```

Syntax

shutdown

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# shutdown
cli:172.16.19.10:interface:cable:csi(1/1/1)# no shutdown
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

sleep

Hides the command prompt for a specified number of seconds.

Mode

Any.

Syntax

sleep <seconds>

Arguments

<seconds>	Specifies the number of seconds to hide the command prompt. Values range from 0 to 4294967295.

Example

```
cli:192.168.208.3:root# sleep 2
sleeping for 2 seconds
cli:192.168.208.3:root#
```

slot

Enters configuration mode for a specified slot.

Mode

Any.

Syntax

slot <c/s>

Arguments

<c/s>	Specify the slot in chassis/slot format.

Example

```
cli:172.16.19.10:root# slot 1/1
mode: slot(1/1)
cli:172.16.19.10:slot(1/1)# show mode
mode: slot(1/1)
cli:172.16.19.10:slot(1/1)# slot 1/8
mode: slot(1/8)
cli:172.16.19.10:slot(1/8)# show mode
mode: slot(1/8)
cli:172.16.19.10:slot(1/8)# slot 1/11
mode: slot(1/11)
cli:172.16.19.10:slot(1/11)# show mode
mode: slot(1/11)
cli:172.16.19.10:slot(1/11)#
```

snmp-server community

Creates an SNMP community. An SNMP community is an organization of hosts such as SNMP managers that require access to MIB objects on a device.

Mode

root

Syntax

snmp-server community <community-name> <security-name>
[**address** <ip-address> [**mask** <ip mask>]] [**context** <context>]
[**storage** {**volatile** | **nonvolatile** | **permanent** | **readonly**}]

Arguments

<community-name>	Name that identifies the SNMP community. An SNMP community name is like a user ID or password that allows hosts to access a device's MIB objects.
<security-name>	Specifies the name of the security group that the community uses. If the group does not already exist, two group entries will be created using the name. These entries correspond to the SNMP version 1 and SNMP version 2c entries for the community.
address <ip-address>	IP address of a host or a range of hosts that are members of the SNMP community. If you do not specify an address, all hosts are allowed access using the community string.
mask <ip mask>	Mask for the IP address. The mask allows you to specify a range of hosts. For example, you can specify an IP address of 220.220.0.0 with a mask of 255.255.0.0. This allows any host from 220.220.0.0 through 220.220.255.255 to access MIB objects in the SNMP community.
context <context>	Name of the SNMP context that is used with the specified community when accessing the security group. Keep in mind that the context is one of the parameters that allows access to a group entry, along with group name, security model, and security level.

storage {**volatile** | **nonvolatile** | **permanent** | **readonly**}

Specifies how the community entry is stored:

- volatile – Entry is stored in volatile memory. The information is lost during a system reboot.
- nonvolatile (default) – Entry is stored in non-volatile memory. The information is not lost during a system reboot.
- permanent – Entry is stored in non-volatile memory. You cannot delete the information but you can make modifications.
- readonly – Entry is stored in non-volatile memory. You cannot delete or modify the information.

Example

```
cli:root# snmp-server community beta build address 192.168.20.12 mask
255.255.255.0 context cuda
cli:192.168.208.3:root# show snmp community

row count: 5

Name        Security Name           Context             Storage
-------------------------------------------------------------
bat         all                                         NonVolatile
beta        build                   cuda                NonVolatile
guitraps    guitraps                                    NonVolatile
private     adc                     adc                 NonVolatile
public      adc                     adc                 NonVolatile
```

Related Commands

- **show snmp community** on page 523

snmp-server contact

Specifies contact information for the device, such as a person's name, e-mail address, and telephone extension. Use the **no snmp-server contact** command to remove the contact name.

Mode

`root`

Syntax

`snmp-server contact` <contact-name>

Arguments

<contact-name>	A text string that specifies contact information for the device, such as a person's name, e-mail address, and telephone extension. The string may contain up to 255 characters. If the string contains spaces, make sure you enclose it in quotes.

Example

```
cli:192.168.220.230:root# snmp-server contact "John Smith Ext. 2456"
cli:192.168.220.230:root#
```

Related Commands

- **no snmp-server contact** on page 293

snmp-server context

Defines an SNMP context. An SNMP context is a collection of management information that an SNMP entity can access. A single management information item can be in more than one context. An SNMP entity potentially has access to many contexts.

Typically, you use contexts as part of an SNMP version 3 network management environment. However, you may also associate contexts with SNMP version 1 and SNMP version 2c communities.

Use the **no snmp-server context** command to remove a context.

Mode

`root`

Syntax

```
snmp-server context <context-name>
[storage {volatile | nonvolatile | permanent | readonly}]
[status {enable | disable}]
```

Arguments

<context-name>	Name that identifies a context.
storage {**volatile** | **nonvolatile** | **permanent** | **readonly**}	Specifies how the context entry is stored: ■ volatile – Entry is stored in volatile memory. The information is lost during a system reboot. ■ nonvolatile (default) – Entry is stored in non-volatile memory. The information is not lost during a system reboot. ■ permanent – Entry is stored in non-volatile memory. You cannot delete the information but you can make modifications. ■ readonly – Entry is stored in non-volatile memory. You cannot delete or modify the information.
status {**enable** | **disable**}	Enables or disables the context. By default, the context is enabled.

Example

```
cli:root# snmp-server context adcw status disable
cli:root# show snmp context

row count: 3

Name        Storage      Status
----------------------- ------------
adc         NonVolatile  Active
adcw        NonVolatile  Not In
                         Service
```

Related Commands

- **no snmp-server context** on page 294
- **show snmp context** on page 525

snmp-server group

Associates SNMP elements, such as a context, with a collection of read, write, and notify views. If you create a group without specifying a read, write, or notify view, the group defaults to the SNMP version 1 default read view.

Mode

root

Syntax

snmp-server group <group-name> {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** <readview-name>] [**write** <writeview-name>] [**notify** <notifyview-name>] [**context** <context-name>] [**storage** {**volatile** | **nonvolatile** | **permanent** | **readonly**}]

Arguments

<group-name>	Name of the group.
{**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}	Specify the following values: ■ A security model for processing SNMP messages for the group. The options are **v1**, **v2c**, or **v3**. ■ The minimum level of security necessary to gain access to the group. The options are: auth – Authenticates a packet through either the HMAC MD5 or SHA algorithms. This is a security level that is higher than noauth but lower than priv. noauth – Authenticates a packet through a string match of the user name. This is the lowest security level. priv – Authenticates a packet through either the HMAC MD5 or SHA algorithms. This type of security also encrypts the packet using the CBC-DES (DES-56) algorithm. This is the highest security level. Note that the auth, noauth, and priv security levels apply to SNMP version 3. The noauth security level applies to SNMP version 1 and SNMP version 2c.
read <readview-name>	Authorizes the group to have read access to the view specified by <readview-name>.

write <writeview-name>	Authorizes the group to have write access to the view specified by <writeview-name>.
notify <notifyview-name>	Authorizes the group to have notify access to the view specified by <notifyview-name>.
context <context-name>	Name of the context associated with the group. The default is the null (or empty) context. Keep in mind that the context is one of the parameters that allows access to a group entry, along with group name, security model, and security level.
storage {**volatile** | **nonvolatile** | **permanent** | **readonly**}	Specifies how the group entry is stored: ■ volatile – Entry is stored in volatile memory. The information is lost during a system reboot. ■ nonvolatile (default) – Entry is stored in non-volatile memory. The information is not lost during a system reboot. ■ permanent – Entry is stored in non-volatile memory. You cannot delete the information but you can make modifications. ■ readonly – Entry is stored in non-volatile memory. You cannot delete or modify the information.

Example

```
cli:192.168.220.230:root# snmp-server group team v3 auth
cli:192.168.220.230:root# show snmp group

row count: 8

Group      Context Model Level  Read View   Write View  Notify View Storage
---------- ------- ----- ------ ----------- ----------- ----------- ---------
adc        adc     V1    NoAuth public      private     public      NonVolatile
adc        adc     V2c   NoAuth public      private     public      NonVolatile
adc        adc     V3    NoAuth public      private     public      NonVolatile
bbb                V1    NoAuth v1default                         NonVolatile
bbb                V2c   NoAuth v1default                         NonVolatile
team               V3    Auth   v1default                         NonVolatile
guitraps           V1    NoAuth                        guitraps    NonVolatile
guitraps           V2c   NoAuth                        guitraps    NonVolatile

cli:192.168.220.230:root# show snmp group team

row count: 1

Group      Context Model Level  Read View   Write View  Notify View Storage
---------- ------- ----- ------ ----------- ----------- ----------- ---------
team               V3    Auth   v1default                         NonVolatile
```

Related Commands

- **show snmp group** on page 527

snmp-server host

Adds a host such as an SNMP manager to the list of notification recipients. Notifications include traps and informs.

Mode

root

Syntax

```
snmp-server host <ip-address> <community-name>
{traps | informs [timeout <seconds>] [retries <retries>]}
[version {1 | 2c | 3 {auth | noauth | priv}}]
[udp-port <port>] [mms <size>]
[storage {volatile | nonvolatile | permanent | readonly}]
[notification-type <type>...]
```

Arguments

<ip-address>	IP address of the SNMP host.
<community-name>	Name of the community that the host uses to receive traps. Note that this name also acts as the security group name. If the group does not already exist, two group entries will be created using the name. These entries correspond to the SNMP version 1 and SNMP version 2c entries for the community.
traps | **informs** [**timeout** <seconds>] [**retries** <retries>]	Specify one of these arguments: ■ traps – Any messages generated contain unconfirmed PDUs. ■ informs – Any messages generated contain confirmed PDUs. Optionally, you can specify the following parameters: timeout <seconds> – Number of seconds that pass before it is assumed the host did not receive the notification message. The range is 0 to 9999 and the default is 15. retries <retries> – Number of retries made when a response to a generated message is not received. The range is 0 to 255 and the default is 3. Note that you cannot use the informs argument if you specify SNMP version 1.

version {**1	2c	3** {**auth	noauth	priv**}}	Specify the following values: ■ A security model for processing SNMP notification messages. The options are 1, 2c, or 3. ■ Specify the minimum level of security necessary for notification message processing. The options are: auth – Authenticates a packet through either the HMAC MD5 or SHA algorithms. noauth – Authenticates a packet through a string match of the user name. priv – Authenticates a packet through either the HMAC MD5 or SHA algorithms. This type of security also encrypts the packet using the CBC-DES (DES-56) algorithm. Note that the auth, noauth, and priv security levels apply to SNMP version 3, for all practical purposes. The noauth security level can be used with SNMP version 1 and SNMP version 2c, if necessary.
udp-port <port>	UDP port to which notification is sent. The range is 0 to 65535 and the default is 162.
mms <size>	Maximum message size (in bytes) of an SNMP message that the SNMP engine transmits or receives and processes. You may specify a value of 0 (no maximum) or a number in the range from 484 to 65535. The default is 484.
storage {**volatile	nonvolatile	permanent	readonly**}	Specifies how the host entry is stored: ■ volatile – Entry is stored in volatile memory. The information is lost during a system reboot. ■ nonvolatile (default) – Entry is stored in non-volatile memory. The information is not lost during a system reboot. ■ permanent – Entry is stored in non-volatile memory. You cannot delete the information but you can make modifications. ■ readonly – Entry is stored in non-volatile memory. You cannot delete or modify the information.
notification-type <type>...	Specify one or more notification types (traps) sent to the host. By default, all types of notifications can be sent to the host. See Table 2-3 for descriptions of notification types. If you do not specify a notification type when you create the host entry, then all notification types are enabled.

Table 2-3 Notification Types

Notification Type	Indicates that...
authentication-failure	The SNMP entity on the Cuda 12000, acting in an agent role, has received a protocol message that is not properly authenticated. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
bcm-failover-down	FastFlow BPM services are going down. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
bcm-failover-up	FastFlow BPM services are coming up. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
bcm-state-change	A change in the IP address configuration of the craft port on the Cuda 12000 has occurred. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
bcm-sw-mismatch	The secondary FastFlow BPM will not come up because its software revision does not match the software revision of the primary. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
cable-modem-auth-failure	An unauthorized cable modem attempted to connect to the system. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
cable-modem-down	A cable modem has gone down. This notification type has a "critical" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
cable-modem-up	A cable modem has come up. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
card-down	A card on the Cuda 12000 has gone down. This notification type has a "critical" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
card-up	A card on the Cuda 12000 has come up. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.

Notification Type	Indicates that...
chassis-fault	A physical fault occurred within the chassis, its modules, its associated fan tray, its power source, or its clock sources. This notification type has a "critical" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
chassis-fault-cleared	A physical fault that occurred within the chassis, its modules its fan tray, its power source, or its clock sources has cleared. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
cold-start	The SNMP entity on the Cuda 12000, acting in an agent role, is reinitializing itself and that its configuration may have been altered. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
dereg-modems	A number or percentage of modems have deregistered over the deregistration time interval. This notification type has a "warning" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
dhcp-relay-not-configured	A DHCP configuration error occurred. This notification type has a "warning" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
docs-bpi-init	A BPI initialization attempt failed during the cable modem registration process and was detected by the CMTS. This notification type has an "info" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
docs-bpkm	A baseline privacy key management (BPKM) operation failed and was detected by the CMTS. This notification type has an "error" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
docs-dcc-ack-fail	A dynamic channel change acknowledgement failed during the dynamic channel change process in the CMTS. This notification type has a "warning" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.

Notification Type	Indicates that...
docs-dcc-req-fail	A dynamic channel change request failed during the dynamic channel change process in the cable modem and was detected by the CMTS. This notification type has a "warning" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
docs-dcc-rsp-fail	A dynamic channel change response failed during the dynamic channel change process in the CMTS. This notification type has a "warning" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
docs-dyn-ack-fail	A dynamic service acknowledgement failed during the dynamic services process and was detected by the CMTS. This notification type has a "warning" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
docs-dyn-req-fail	A dynamic service request failed during the dynamic services process and was detected by the CMTS. This notification type has a "warning" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
docs-dyn-rsp-fail	A dynamic service response failed during the dynamic services process and was detected by the CMTS. This notification type has a "warning" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
docs-dynamic-sa	A dynamic security association operation failed and was detected by the CMTS. This notification type has an "error" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
docs-init-ack-fail	A registration acknowledgement from a cable modem failed during the cable modem initialization process and was detected by the CMTS. This notification type has a "warning" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
docs-init-req-fail	A registration request from a cable modem failed during the cable modem initialization process and was detected by the CMTS. This notification type has a "warning" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.

Notification Type	Indicates that...
docs-init-rsp-fail	A registration response failed during the cable modem initialization process and was detected by the CMTS. This notification type has a "warning" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
duplicate-addr	A duplicate IP address has been detected. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
icl-state-change	A change in the ICL link to a peer has occurred. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
isp-addr-high	The free IP address count exceeded the upper threshold for the specified ISP. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
isp-addr-low	The free address count fell below the lower threshold for the specified ISP. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
ldap-failed	A directory server access failure occurred. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
ldap-restored	Directory server access is operational after a failure. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
link-down	Signifies that the SNMP entity on the Cuda 12000, acting in an agent role, has detected that the communication link is about to go down. This notification type has an "error" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
link-up	Signifies that the SNMP entity on the Cuda 12000, acting in an agent role, has detected that the communication link is about to come up. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.

Notification Type	Indicates that...
local-sonet-alarm	A SONET local alarm is occurring. This notification type has an "error" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
ospf-nbr-state	Signifies a change in the state of an OSPF neighbor on a physical interface. This notification type has a "notice" priority. To send this notification type, note that you also have to enable sending of OSPF neighbor state traps using the **report** command. Refer to the description of the **report** command on page 348 for more information.
ospf-virt-nbr-state	Signifies a change in the state of an OSPF neighbor on a virtual interface. This notification type has a "notice" priority. To send this notification type, note that you also have to enable sending of OSPF virtual neighbor state traps using the **report** command. Refer to the description of the **report** command on page 348 for more information.
prov-service	A FastFlow BPM service started, stopped, or failed. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
remote-sonet-alarm	A SONET remote alarm is occurring. This notification type has an "error" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
subnet-addr-high	The free IP address count exceeded the high available address threshold for a subnet. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
subnet-addr-low	The free IP address count fell below the low available address threshold for a subnet. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
trace-log	For ADC internal use only. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.
warm-start	Signifies that the SNMP entity on the Cuda 12000, acting in an agent role, is reinitializing itself such that its configuration is unaltered. This notification type has a "notice" priority. Refer to the description of the **event-config reporting** command for more information on event priorities.

Example

```
cli:192.168.220.230:root# snmp-server host 136.4.6.6 private informs timeout
500 retries 200 version 2c udp-port 164 mms 5000 notification-type cold-start
link-up
cli:192.168.220.230:root#
```

Related Commands

- **report** on page 348
- **show snmp host** on page 528

snmp-server location

Specifies the physical location of the device. Use the **no snmp-server location** command to remove the location string.

Mode

```
root
```

Syntax

```
snmp-server location <location>
```

Arguments

<location>	A text string that describes the physical location of the device (sysLocation MIB object). The string may contain up to 255 characters. If the string contains spaces, enclose it in quotes.

Example

```
cli:192.168.220.230:root# snmp-server location "Building 2, First Floor"
cli:192.168.220.230:root#
```

Related Commands

- **no snmp-server location** on page 299

snmp-server name

Specifies the system name (sysName) of the device. Use the **no snmp-server name** command to remove the name.

Mode

`root`

Syntax

`snmp-server name` <sysname>

Arguments

<sysname>	A text string that specifies the system name (sysName MIB object). The name may contain up to 255 characters.

Example

```
cli:192.168.220.230:root# snmp-server name cuda1
cli:192.168.220.230:root#
```

Related Commands

- **no snmp-server name** on page 300

snmp-server user

Creates an SNMP user account.

Mode

root

Syntax

```
snmp-server user <user> [auth {md5 | sha} <auth-password>
[priv des56 <priv-password>]]
[storage {volatile | nonvolatile | permanent | readonly}]
[status {enable | disable}]
```

Arguments

<user>

Name of the user. Range is 1 to 32 characters and may not contain spaces.

auth {**md5** | **sha**} <auth-password> [**priv des56** <priv-password>]

You must specify the following arguments:

- auth – Specify one of these authentication types:

  md5 – HMAC-MD5-96

  sha – HMAC-SHA-96

- <auth-password> – Password key for authentication. If the authentication type is HMAC-MD5-96 or HMAC-SHA-96, you must enter a password for authentication. The password must be an ASCII Hex string with a maximum size of 32 characters (16 bytes) for HMAC-MD5-96 and 40 characters (20 bytes) for HMAC-SHA-96.

  Password is write-only for security purposes. You must enter the password each time you make a modification.

Optionally, you may use the priv des56 argument, which specifies a password (<priv-password>) that is encrypted with the DES algorithm. This encryption protects messages from disclosure. The password must be an ASCII Hex string with a maximum size of 64 characters.

storage {**volatile** | **nonvolatile** | **permanent** | **readonly**}	Specifies how the user entry is stored: ■ volatile – Entry is stored in volatile memory. The information is lost during a system reboot. ■ nonvolatile (default) – Entry is stored in non-volatile memory. The information is not lost during a system reboot. ■ permanent – Entry is stored in non-volatile memory. You cannot delete the information but you can make modifications. ■ readonly – Entry is stored in non-volatile memory. You cannot delete or modify the information.
status {**enable** | **disable**}	Enables or disables the user account. The default is enabled.

Example

```
cli:192.168.220.207:root# snmp-server user dave_jones auth md5 ab24fdc1
cli:192.168.220.207:root#
```

Related Commands

- **no snmp-server user** on page 301
- **show snmp user** on page 532

snmp-server view

Creates an SNMP access view, which controls access to a MIB subtree.

Mode

root

Syntax

```
snmp-server view <view-name> <oid-tree>
{included | excluded}
[storage {volatile | nonvolatile | permanent | readonly}]
[status {enable | disable}]
```

Arguments

<view-name>	Name of the view. If the name is a zero length string, no access is allowed. The range is 1 to 32 characters.
<oid-tree>	MIB subtree that defines the family of view subtrees. You can enter in the MIB value as an Object Identifier (OID), an OID with wildcards, or an OID name description, such as, sysDescr.
{**included** | **excluded**}	Specify one of these arguments: ■ included – The corresponding instances of the MIB subtree are included in the MIB view. ■ excluded – The corresponding instances of the MIB subtree are excluded from the MIB view.
storage {**volatile** | **nonvolatile** | **permanent** | **readonly**}	Specifies how the view entry is stored: ■ volatile – Entry is stored in volatile memory. The information is lost during a system reboot. ■ nonvolatile (default) – Entry is stored in non-volatile memory. The information is not lost during a system reboot. ■ permanent – Entry is stored in non-volatile memory. You cannot delete the information but you can make modifications. ■ readonly – Entry is stored in non-volatile memory. You cannot delete or modify the information.
status {**enable** | **disable**}	Enables or disables the view entry. The entry is enabled by default.

Example 1

In this example, the user allows access to the entire MIB:

```
cli:192.168.208.3:root# snmp-server view auditorview1 1.3.6.1 included
cli:192.168.208.3:root# show snmp view

row count: 5

View Name        Subtree                     Type     Storage     Status
---------------- --------------------------- -------- ----------- -----------
public           1.3.6.1                     Included NonVolatile Active
private          1.3.6.1                     Included NonVolatile Active
guitraps         1.3.6.1                     Included NonVolatile Active
v1default        1.3.6.1                     Included NonVolatile Active
auditorview1     1.3.6.1                     Included NonVolatile Active
```

Example 2

In this example, the user excludes sensitive SNMP configuration information:

```
cli:192.168.208.3:root# snmp-server view auditorview2 snmpModules excluded
cli:192.168.208.3:root# show snmp view

row count: 5

View Name        Subtree                     Type     Storage     Status
---------------- --------------------------- -------- ----------- -----------
public           1.3.6.1                     Included NonVolatile Active
private          1.3.6.1                     Included NonVolatile Active
guitraps         1.3.6.1                     Included ReadOnly    Active
v1default        1.3.6.1                     Included NonVolatile Active
auditorview2     1.3.6.1.6.3                 Excluded NonVolatile Active
```

Example 3

In this example, the user excludes interface table (ifTable) entries for interface (ifIndex) 8781825:

```
cli:root# snmp-server view auditorview3 1.3.6.2.1.2.2.*.8781825 excluded
cli:root# show snmp view

row count: 5

View Name        Subtree                      Type     Storage     Status
---------------- ---------------------------- -------- ----------- -----------
public           1.3.6.1                      Included NonVolatile Active
private          1.3.6.1                      Included NonVolatile Active
guitraps         1.3.6.1                      Included NonVolatile Active
v1default        1.3.6.1                      Included NonVolatile Active
auditorview3     1.3.6.2.1.2.2.*.8781825      Excluded NonVolatile Active
```

Related Commands

- **no snmp-server view** on page 302
- **show snmp view** on page 533

spectrum-group

Configures or modifies a frequency hopping policy on an upstream channel. Use the no form of the command to remove the policy from the upstream channel.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
spectrum-group <rule-number> upstream <channel-id>
profile <upstream-modulation-number> threshold <number>
interval <number> frequency <number>

no spectrum-group <rule-number> upstream <channel-id>
```

Arguments

<rule-number>	Specifies the user-defined number assigned to a policy configuration. Five policies may be configured for each upstream channel. Values range from 1 to 5.
upstream <channel-id>	Specifies the upstream channel port on which the policy set. Valid range is 1 to 4 (1x4 modules) and 1 to 6 (1x6 modules).
profile <upstream-modulation-number>	Specifies the modulation profile number to which the policy configuration is set.
threshold <number>	Specifies the percentage error threshold for this frequency hopping policy entry. Values range from 1 to 100. For example, if you specify 50, then the error threshold is reached if errors occur 50 percent of the time during the threshold interval.
interval <number>	Specifies the threshold interval for this frequency hopping policy entry in seconds. Values range from 10 to 86400.
frequency <number>	Specifies the center frequency value to be used when the error threshold is reached in the configured threshold interval time. Values range from 5.0 to 42.0.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# spectrum-group 2 upstream 2
profile 1 threshold 1 interval 86400 frequency 42.0
```

Related Commands

- **show spectrum-group** on page 535
- **spectrum-group reset upstream** on page 581

spectrum-group reset upstream

Resets an upstream channel back to policy 1.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
spectrum-group reset upstream <channel-id>
```

Arguments

<channel-id>	Specifies the ID of the upstream channel that you want to set back to policy 1. Values range from 1 to 4 for 1x4 modules and 1 to 6 for 1x6 modules. Use the **show spectrum-group** command to display upstream channel IDs.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# spectrum-group reset upstream 1
cli:192.168.208.3:interface:cable:csi(1/1/1)#
```

Related Commands

- **show spectrum-group** on page 535
- **spectrum-group** on page 579

speed

Sets the speed on an Ethernet port.

Mode

```
interface:ethernet:csi(<c/s/i>)
```

Syntax

speed {**auto** | **10** | **100**}

Arguments

{**auto** | **10** | **100**} Specifies one of the following values:

- auto – Speed is automatically negotiated. Note that, if you configure the port to automatically negotiate speed, you also set the port to automatically negotiate duplex mode.
- 10 – Sets the speed to 10 Mbps. Note that, if you configure the port for 10 Mbps or 100 Mbps, you also set the duplex mode to it's last explicit setting (half or full).
- 100 – Sets the speed to 100 Mbps.

Example

```
cli:192.168.208.3:interface:ethernet:csi(1/11/1)# speed 100
cli:192.168.208.3:interface:ethernet:csi(1/11/1)# show interface
Interface                  1/11/1 Ethernet (100 Mb)
Configured duplex                        full
Configured speed                       100 Mb
Auto-negotiation                      disable

In octets                                   0
In unicast                                  0
In multicast                                0
In broadcast                                0

Out octets                                  0
Out unicast                                 0
Out multicast                               0
Out broadcast                               0
```

Related Commands

- **duplex** on page 156
- **negotiation auto** on page 259
- **show interface** on page 414

sync-interval

Specifies the time interval between SYNC message transmission on the downstream port.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

sync-interval <value>

Arguments

<value>	Time, in milliseconds, between downstream SYNC message transmission. Range: 1 to 200 milliseconds. Default: 5 milliseconds.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# sync-interval 200
cli:172.16.19.10:interface:cable:csi(1/1/1)# show mac | include "Sync Interval"

Sync Interval                           200  (millisec)
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421
- **show mac** on page 454

tacacs-server

This command sets the IP address and encryption key of the TACACS+ authentication server.

Mode

`root`

Syntax

`tacacs-server` {**`host`** <ip-address> | **`key`** <string>}

Arguments

host <ip address>	The IP address of the host TACACS+ authentication server.
key <string>	The encryption key of the host TACACS+ authentication server.

Example 1

This example illustrates how to set the IP address of the host TACACS+ authentication server.

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# root
mode: root
cli:192.168.208.3:root# tacacs-server host 192.168.208.3
cli:192.168.208.3:root#
```

Example 2

This example illustrates how to set the encryption key of the host TACACS+ authentication server.

```
cli:192.168.208.3:root# tacacs-server key cuda12000
cli:192.168.208.3:root#
```

Related Commands

- **aaa authentication** on page 44
- **show aaa** on page 368
- **show tacacs-server** on page 540

talk

Allows you to perform the following tasks:

- Enable and disable the ability to broadcast messages to all users
- Send broadcast messages to all users

Mode

Any.

Syntax

```
talk {on | off | history | <message-string>}
```

Arguments

{**on | off | history |** <message-string>}	Specifies one of the following values: ■ on – Enables sending of broadcast messages to all Cuda 12000 users. ■ off – Disables sending of broadcast messages to all Cuda 12000 users. ■ history – Displays a history of broadcast messages. ■ <message-string> – A broadcast message string. If the string contains spaces, enclose it in quotes. Question marks are not allowed.

Example

```
cli:root# talk "System shutdown in 5 minutes."
Sending  message: System shutdown in 5 minutes.

Received message: System shutdown in 5 minutes.
```

trace-log

Trace log allows you to trace specific information through basmonitor. Tracing allows you to monitor and view specific information; you can use this information for performance monitoring, troubleshooting, and debugging purposes.

The **trace-log** command defines the information that basmonitor displays. You can use this command to configure viewing within the following information areas:

- **Slot Specific** — You can configure trace log information from with slot(<c/s>) mode to view and monitor information on a slot-wide basis. For example, you can configure the tracing of DHCP information that is relayed over the slot, or widen the scope to IP specific data. Within slot mode, the steps to configure trace-log include:

  1. Specifying the information *(*in terms of software component*)* that you want to monitor.
  2. Configuring the log level for that information.

- **Cable Modem Specific** — You can configure trace log information from within interface:cable:csi(<c/s/i>) mode to view and monitor information as it pertains to registration of a single cable modem. This allows you to view and monitor the information flow that occurs during a registration of a specific cable modem; use this information to debug registration failures on a per cable modem basis. Within cable interface mode, the steps to configure trace-log include:

  1. Specify the cable modem that you want to monitor.
  2. Configure whether you want to monitor the cable modem ranging messages.
  3. Configure whether you want to monitor cable registration messages.
  4. Configure whether you want to monitor cable modem baseline privacy messages.
  5. Configure the log-level at which you want to monitor the specified information.

The information that you can configure to view in basmonitor using the **trace-log** command depends on the mode in which you launch it. The log-level controls how much of the specified information is displayed and controls the scope of the information.

Mode

To monitor and display trace information on a slot-wide basis, execute the **trace-log** command in the following mode:

`slot(c/s)`

To monitor and display trace information on a CMTS interface basis, execute the **trace-log** command in the following mode:

`interface:cable:csi(<c/s/i>)`

Syntax

Within slot(c/i) mode:

`trace-log` `<sw-component-id> <log-level>`

Within interface:cable:csi(<c/s/i>) mode, all trace-log configuration is performed in an effort to monitor specific registration process information for a particular modem.

Specify the cable modem that you want to monitor:

`trace-log cm mac-address` `<mac-address>`

Configure whether you want to monitor the cable modem ranging messages:

`trace-log ranging {true | false}`

Configure whether you want to monitor cable registration messages:

`trace-log registration {true | false}`

Configure whether you want to monitor cable modem baseline privacy messages:

`trace-log baseline-privacy {true | false}`

Configure the log-level at which you want to monitor the specified information:

`trace-log detail-level {highest | high | medium | low}`

Arguments

Within slot mode:

<sw-component-id>	The ID of the software component that you want to monitor. These IDs are listed in Table 2-4.
<log-level>	The software component's trace logging severity level. One of the following: ■ highest ■ high ■ medium ■ low

Within cable interface mode:

cm mac-address <cm-mac-address>	Specifies the MAC address of the modem that you want to monitor.
ranging {**true** | **false**}	Enables and disables the monitoring of messages that occur during the ranging stage.
registration {**true** | **false**}	Enables and disables the monitoring of messages that occur during the registration stage.
baseline-privacy {**true** | **false**}	Enables and disables the monitoring of messages that occur during the baseline privacy stage (applicable only when baseline privacy is enabled).
detail-level {**highest** | **high** | **medium** | **low**}	Specifies the amount of detail: highest, high, medium, or low.

The following table lists the software component IDs that you can specify for monitoring within slot mode:



Debug commands are for ADC use only.

Table 2-4 Software Component IDs

agentx – Events related to the agentx protocol between the master SNMP agent on the BCM and the and local agents on the I/O modules.	**basctlr** – Events related to the management controller. Only valid on slots 13 and 14.
ca – SNMP issues on the BCM.	**cfg-rmi** – Communications issues related to the configuration process.
cfm – Configuration File Manager.	**cli** – Command Line Interface events. Only valid on slots 13 and 14.
cmts – CMTS events other than MAC and BPI.	**cmts**-<0-17> – CMTS event levels 0 through 17. For example: **cmts-0**. For ADC developer debug use only.
cmts_bpi – Baseline Privacy events.	**cmts-docsis-err** – DOCSIS specific errors.
cmts-generic – General CMTS events.	**cmts-mac** – MAC only events.
courier – Courier events.	**crp** – Not used.
dhcp-relay – Events related to DHCP on an application module.	**dhcp-server** – DHCP Server events. Only valid on slots 13 and 14.
dns-server – DNS Server events. Only valid on slots 13 and 14.	**faults-rmi** – Errors related to processing alarms.
ftd – Events related to distributing the IP forwarding table.	**ha-interface** – Errors related to the high availability (HA) interface. This interface is part of support for fault tolerant redundancy.
icmp – ICMP events.	**idle** – The idle task.
igmp – IGMP events.	**ip** – IP forwarding events.
ip-debug – Extensive debug of IP, provides function level debugging. For ADC developer debug use only.	**java-server** – Java server supporting CudaView and CLI.
jni – Events related to the Java Network Interface, which is required for multi-chassis support.	**la** – SNMP local agent on slots 1 – 12.
ldap-client – Events related to the LDAP client.	**ldp** – Link Discover Protocol on slots 1 - 12.
log – log task.	**ma** – SNMP Master agent (slot 13 or 14 only).
mal – Management Access Layer which supports both CLI and CudaView.	**mcast** – Multicast-related events.
mtd – MTD-related events.	**nlbg** – Bridge-group events.

Table 2-4 Software Component IDs

nlbg-cmts – CMTS-related bridge-group events.	**nlbg-rs** – Route-server-related bridge-group events.
ospf – OSPF protocol events.	**ospf-debug** – OSPF debug events. For ADC developer debug use only.
ospf-hello – OSPF hello packet events.	**ospf-spf** – OSPF SPF events.
ppp – PPP events.	**ppp-debug** – PPP debug events. For ADC developer debug use only.
prov-rmi – Errors related to the provisioning server/CudaView interaction.	**rbp** – Events related to the ADC proprietary reliable IPC mechanism.
rcv – The receive module within the switching layer.	**relay-to-syslog** – Events related to relaying messages to the SYSLOG server.
rip – Not used.	**rip-console** – Extensive debug of RIP, provides function level debugging. For ADC developer debug use only.
rip-debug – RIP debug events. For ADC developer debug use only.	**rip-general** – Miscellaneous RIP events.
rip_mem – Events related to RIP memory allocation.	**rip-route** – Events related to route updating.
rip-rx – Events occurring during RIP receive packet handling.	**rip-state** – Events related to internal RIP states.
rip-task – Events which occur within the main RIP task.	**rip-timer** – Events related to RIP periodic timers.
rip-tx – Events occurring during RIP transmission.	**rm** – Resource Manager events.
routing-trace – Trap based debugging for all routing protocols. Misconfigurations would be caught here.	**short-msg** – Short message events.
snmp – Events related to the SNMP protocol.	**sw** – Switching Layer events.
switching-tftpd – Switching events related to the TFTP server.	**syslog-server** – SYSLOG Server events. Only valid on slots 13 and 14.
tcp – TCP events.	**tftp-server** – TFTP Server events. Only valid on slots 13 and 14.
time-server – Time Server events. Only valid on slots 13 and 14.	**tracelogd** – Tracelog Server events. Only valid on slots 13 and 14.
udp – UDP events.	

Example

The following example uses **trace-log** to configure basmonitor to display critical messages exchanged between cable modem 00:90:96:00:29:6d and the DHCP server over cable interface 1/1/1. In the same example, the monitoring of baseline privacy exchanges is disabled. After trace-log configuration, the specified cable modem is reset and basmonitor is launched to view the DOCSIS initialization messages upon cable modem bootup.

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# show modem 00:90:96:00:29:6d

row count: 1

MAC Address       IP Address      SID  Down   Up     Power  Timing Modem State
                                       Stream Stream (dbMV)
----------------- --------------- ---- ------ ------ ------ ------ ------------
00:90:96:00:29:6d 192.168.19.52    662      1      1      0   1129 Registered

cli:172.16.19.10:interface:cable:csi(1/1/1)# trace-log cm mac-address 00:90:96:00:29:6d
cli:172.16.19.10:interface:cable:csi(1/1/1)# trace-log log-level critical
cli:172.16.19.10:interface:cable:csi(1/1/1)# trace-log ranging true
cli:172.16.19.10:interface:cable:csi(1/1/1)# trace-log baseline-privacy false
cli:172.16.19.10:interface:cable:csi(1/1/1)# trace-log registration true
cli:172.16.19.10:interface:cable:csi(1/1/1)# show trace-log
Mac Address                00:90:96:00:29:6d
Log Level                          critical
Ranging                             True
Registration                        True
Baseline-Privacy                   False

cli:172.16.19.10:interface:cable:csi(1/1/1)# cm reset 00:90:96:00:29:6d
Reseting Cable Modem4300 with Mac Address: 00:90:96:00:29:6d
cli:172.16.19.10:interface:cable:csi(1/1/1)# basmonitor

1 / 1 / 0 / 2 - DHCP Relay - info 14:30:12: (REQUEST)
1 / 1 / 0 / 2 - DHCP Relay - info 14:30:12: dhcp_relay: cable modem 00:90:96:00:
29:6d.
1 / 1 / 0 / 2 - DHCP Relay - info 14:30:12: add_relay_agent_options: added circu
it id (16811520) len 4 for cm
1 / 1 / 0 / 2 - DHCP Relay - info 14:30:12: add_relay_agent_options: added rem i
d (00:90:96:00:29:6d) len 6
1 / 1 / 0 / 2 - DHCP Relay - info 14:30:12: dhcp_relay: sent REQ to server 10.1.
13.1 DHCP relay 192.168.19.50
1 / 1 / 0 / 2 - DHCP Relay - info 14:30:12: (ACK)
1 / 1 / 0 / 2 - DHCP Relay - info 14:30:12: dhcrelay: sending unicast to 00:90:9
6:00:29:6d
1 / 1 / 0 / 2 - DHCP Relay - info 14:30:12: dhcrelay(): calling msg_send_remote
to 10.1.1.2
1 / 1 / 0 / 2 - DHCP Relay - info 14:30:12: strip_relay_agent_options: interface
 found circuit id (16811520).
1 / 1 / 0 / 2 - DHCP Relay - info 14:30:12: dhcp_relay: forwarded BOOTREPLY for
00:90:96:00:29:6d to 192.168.19.52
```

traceroute

Traces the route that packets traverse from the Cuda 12000 to a destination. All arguments except the destination IP address are optional.

Consider using this command in conjunction with the **ping** command. For example, you can use the **ping** command to determine if a host is reachable. You then can use the **traceroute** command to determine the route to the host.

Mode

Any.

Syntax

```
traceroute [-w <timeout>] [-i <number>] [-m <number>]
[-p <number>][-q <number>] [-t <number>] [-s <src-ip-address>]
[-x <number>] [-F] <dest-ip-address> [<data-size>]
```

Arguments

-w <timeout>	Specifies the number of seconds to wait for a response to a probe packet. Values range from 1 to 30. The default is 1. Note that a trace consists of a series of transmitted probe packets.
-i <number>	Specifies the initial time-to-live (TTL) value in number of hops, enabling you to bypass the initial (often well known) portion of a path. Using this argument, you can configure the trace to ignore hosts that are less than the specified number of hops away from your Cuda 12000. Values range from 0 (no initial TTL) to 255. The default is 1.
-m <number>	Specifies the maximum TTL value in number of hops. When the Cuda 12000 sends a traceroute packet, the Cuda 12000 sets the TTL value in the packet to the value you specify. Each time a router forwards the packet, the router decrements this value by one. Routers discard packets that have a TTL of zero. Values range from 1 to 255. The default is 30.

-p <number>	Specifies the base UDP port number on the destination host to which the traceroute command sends probe packets. Values range from 1 to 65535. The default is 33434. The traceroute command assumes that no other processes on the destination host use UDP port numbers in the range of base to base + nhops -1. For example, if the base is 33434, then the traceroute command uses a UDP port in the range: 33434 to 33434 + nhops - 1 If another process listens on a port in this range, you can use this argument to specify a new base UDP port number, thereby configuring an unused port range.
-q <number>	Specifies the number of probe packets sent to each hop. Values range from 1 to 10. The default is 3.
-t <number>	Specifies the type-of-service (ToS) value in probe packets. Use this argument to determine if different ToS take different paths. Useful values are 16 (low delay) and 8 (high throughput). The value must be a decimal integer in the range from 0 to 255. The default is 0.
-s <src-ip-address>	Specifies the IP address (which you must supply as an IP number, not a hostname) on your Cuda 12000 that will be the source address in outgoing probe packets. By default, the source address is the IP address of the interface on which the Cuda 12000 sends the probe packet. If your Cuda 12000 has more than one IP address, this argument enables you to override the default source address.
-x <number>	Specifies the maximum number of consecutive timeouts. The traceroute command stops the trace when this threshold is reached. Values range from 0 to 255. The default is 5.
-F	Disables IP fragmentation for the trace. If you disable fragmentation, and the packet size that you specify with the <data-size> argument is so big that routers fragment it along the route, the traceroute command indicates that fragmentation has occurred. If a router returns the value of the proper MTU size, traceroute decreases the packet size automatically to this new value. Otherwise, traceroute chooses a shorter packet size. By default, fragmentation is enabled. You must specify the -F argument to disable it.
<dest-ip-address>	Specifies the destination IP address for the trace.
<data-size>	Specifies the size, in bytes, of the probe packets in the trace. Values range from 64 bytes to 64000 bytes. The default is 64 bytes.

Example

```
cli:192.168.220.230:root# traceroute -w 3 -q 5 -x 10 216.52.0.12 1000
traceroute to (216.52.0.12), 30 hops max, 1000 byte packets
Hop #1: !!!!! (205.171.30.85) Success rate is 100.0 percent (5/5)
Hop #2: !.!!! (205.171.5.115) Success rate is 80.0 percent (4/5)
Hop #3: !!!!! (205.171.5.121) Success rate is 100.0 percent (5/5)
Hop #4: !!!!! (205.171.22.30) Success rate is 100.0 percent (5/5)
Hop #5: !!..! (208.46.223.34) Success rate is 60.0 percent (3/5)
Hop #6: !!!!! (216.52.0.12) Success rate is 100.0 percent (5/5)
```

Related Commands

- **ping** on page 312

traffic-relay

Configures processes, such as the HTTP server, to send and receive TCP or UDP packets using an internal address on the Cuda 12000. This method of sending and receiving packets is called traffic relay.

If you are running a TFTP server on the Cuda 12000 as part of FastFlow BPM provisioning, you must enable traffic relay for the TFTP server in order to download configuration files to cable modems. The TFTP server sends and receives packets using an internal address. Refer to the FastFlow BPM documentation set for more information on the FastFlow BPM.

The **traffic-relay** command also allows you to configure the Cuda 12000 for in-band management. For example, you can use this command to enable forwarding of Telnet traffic and HTTP traffic using an internal address, thereby allowing you to perform in-band management of the Cuda 12000 using the CLI or CudaView.

Use the no form of the command to disable traffic relay for a server.

Mode

`root`

Syntax

```
traffic-relay {dns | ftp | http | snmp | snmp-trap | ssh | syslog |
  telnet | tftp | time_of_day} [port <port>]

no traffic-relay {dns | ftp | http | snmp | snmp-trap | ssh | syslog |
  telnet | tftp | time_of_day}
```

Arguments

dns	Enables traffic relay for the DNS server. The no form of the command disables traffic relay for the DNS server.
ftp	Enables traffic relay for FTP communications. The no form of the command disables traffic relay for FTP communications.
http	Enables traffic relay for the HTTP server. The no form of the command disables traffic relay for the HTTP server.
snmp	Enables traffic relay for the SNMP agent. The no form of the command disables traffic relay for the SNMP agent.
snmp-trap	Enables traffic-relay for the sending of SNMP traps. The no form of the command disables traffic-relay for the sending of SNMP traps.

ssh	Enables traffic relay for the secure shell server. The no form of the command disables traffic relay for the secure shell server.
syslog	Enables traffic relay for the syslog server. The no form of the command disables traffic relay for the syslog server.
telnet	Enables traffic relay for the Telnet server. The no form of the command disables traffic relay for the Telnet server.
tftp	Enables traffic relay for the TFTP server. The no form of the command disables traffic relay for the TFTP server.
time_of_day	Enables traffic relay for the Time-of-Day server. The no form of the command disables traffic relay for the Time-of-Day server.
port <port>	Specifies a TCP or UDP port to be used by the specified process for traffic relay. For example, if the HTTP server on the Cuda 12000 uses a port other than the default (80), you can use this argument to specify the non-default port.

Example

```
cli:192.168.208.3:root# traffic-relay dns port 55
cli:192.168.208.3:root# show traffic-relay

row count: 10

Protocol     State    Port Number
------------ -------- -----------
tftp         enable            69
time_of_day  enable            37
syslog       enable           514
dns          enable            55
snmp         disable          161
telnet       disable           23
ssh          disable           22
http         enable            80
ftp          disable           21
snmp-trap    enable           162

cli:192.168.208.3:root#
```

Related Commands

- **show traffic-relay** on page 545

ucd-interval

Use this command to set the time interval between sending successive Upstream Channel Descriptor messages for each upstream channel. The UCD interval specifies how frequently an Upstream Channel Descriptor message is transmitted on every active upstream.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

ucd-interval <value>

Arguments

<value>	Time interval in milliseconds. The valid range is 1 to 2000. The default is 2000 milliseconds.

Example

```
cli:172.16.19.10:root# interface cable 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# ucd-interval 1000
cli:172.16.19.10:interface:cable:csi(1/1/1)# show mac | include UCD
UCD Interval                          1000  (millisec)
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421
- **show mac** on page 454

up

Use this command to back up one level from the current mode.

Mode

Any.

Syntax

up

Example

```
cli:172.16.19.10:root# prov-server
mode: prov-server
cli:172.16.19.10:prov-server# dhcp-server
mode: prov-server:dhcp-server
cli:172.16.19.10:prov-server:dhcp-server# up
mode: prov-server
cli:172.16.19.10:prov-server# up
mode: root
cli:172.16.19.10:root#
```

Related Commands

- **show mode** on page 458

upstream <port number> channel-width

Sets the upstream channel width in kilohertz (kHz).

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

upstream <port number> **channel-width** {**200** | **400** | **800** | **1600** | **3200**}

Arguments

<port number>	Number of upstream port for which you want to set the channel width. Valid range is 1 to 4 for 1x4 modules and 1 to 6 for 1x6 modules.
channel-width {**200** | **400** | **800** | **1600** | **3200**}	Specifies the upstream channel width in kilohertz (kHz). By default, the Channel Width is set at 3200 kHz (2560 kilosymbols - ksyms). Acceptable values include: ■ 200K (160 ksyms per second) ■ 400K (320 ksyms per second) ■ 800K (640 ksyms per second) ■ 1600K (1280 ksyms per second) ■ 3200K (2560 ksyms per second)

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 channel-width 800
cli:172.16.19.10:interface:cable:csi(1/1/1)# show upstream 1 | include Width
Channel Width                              0.8  (MHz)
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421

upstream <port number> data-backoff

Sets a data backoff range on the specified upstream port. By default, the data backoff range is set to 5. Acceptable values are 0 to 15.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
upstream <port number> data-backoff <start-value> <end-value>
```

Arguments

<port number>	Number of upstream port for which you want to set the data backoff range. Valid range is 1 to 4 for 1x4 modules and 1 to 6 for 1x6 modules.
<start-value>	Start of the data backoff range. Valid numbers: 0 –15.
<end-value>	End of the data backoff range. Valid numbers: 0 –15.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 data-backoff 5 10
cli:172.16.19.10:interface:cable:csi(1/1/1)# show upstream 1 | include "Tx Back"
Tx Backoff Start                          5
Tx Backoff End                           10
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421
- **show mac** on page 454

upstream <port number> frequency

Sets the signal frequency (center frequency) of the specified upstream port.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
upstream <port number> frequency <value>
```

Arguments

<port number>	Number of the upstream port that you want to configure. Valid range is 1 to 4 for 1x4 modules and 1 to 6 for 1x6 modules.
frequency <value>	Frequency to which you want to set the upstream port. Valid DOCSIS range: 5.0 - 42.0 MHz. Valid EuroDOCSIS range: 5.0 - 65.0 MHz.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 frequency 20
cli:172.16.19.10:interface:cable:csi(1/1/1)# show upstream 1 | include Center
Center Frequency                        20.0  (MHz)
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421
- **show mac** on page 454

upstream <port number> map

Configures MAP generation for the specified upstream channel.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
upstream <port number> map {init-maint-size-adjust <value> |
  max-ranging-invitations <value> | min-req-region <value> |
  ucd-grant-size <value>}
```

Arguments

<port number>	Number of the upstream port that you want to configure. Valid range is 1 to 4 for 1x4 modules and 1 to 6 for 1x6 modules.
init-maint-size-adjust <value>	The size of the upstream channel Initial Maintenance (IM) contention region. Maps with Initial Maint regions are sent periodically. By default, Initial Maint Contention Region Size is set at 500. Acceptable values: 0-65535.
max-ranging-invitations <value>	Maximum number of deferred ranging invitations. By default, Maximum Deferred Ranging Invitations is set at 2. Acceptable values: 0-65535.
min-req-region <value>	The minimum size, in minislots, for request contention region. By default, Minimum Request Region Size is set at 20.
ucd-grant-size <value>	Upstream Channel Description (UCD) grant size. After a UCD change, this specifies the size of grant to zero (which functions as a delay for cable modems to digest the new UCD), in microseconds. By default, New UCD Grant Size is set at 3000. Acceptable values: 0-65535.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# upstream 1 map
init-maint-size-adjust 10
cli:192.168.208.3:interface:cable:csi(1/1/1)# upstream 1 map
max-ranging-invitations 100
cli:192.168.208.3:interface:cable:csi(1/1/1)# show upstream 1
Upstream Channel Id        1 (1 / 1 / 3 / 2)
Center Frequency                      20.0  (MHz)
Channel Width                       1600.0  (KHz)
Slot Size                                4  (uSec)
Receive Power                            0  (TenthdBmV)
Voice BW Reservation                  75.0  (%)
Modulation Profile                       1
Tx Timing Offset                      1979
Tx Backoff Start                         5
Tx Backoff End                          10
Ranging Backoff Start                    2
Ranging Backoff End                      3

Admin Status                            up
Operational Status                      up

Stats:
In Octets                           617780
In Unicast Packets                   18170
In Multicast Packets                     0
In Broadcast Packets                     0
In Errors                               85
In Discards                              0
--More--
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421
- **show upstream** on page 547

upstream <port number> minislot-size

Specifies the number of 6.25 microsecond ticks in each upstream minislot. This value depends on one selected channel width, which is automatically set when the user selects an acceptable channel width. By default, the minislot size is set at 2.

The following minislot sizes for different channel widths are recommended:

- 2 (3200 kHz)
- 4 (1600 kHz)
- 8 (800 kHz)
- 16 (400 kHz)
- 32 (200 kHz)



The slot size affects the performance of the CMTS. It is recommended that configuration be performed by an expert-level administrator.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

upstream <port number> **minislot-size** <value>

Arguments

<port number>	Number of the upstream port that you want to configure. Valid range is 1 to 4 for 1x4 modules and 1 to 6 for 1x6 modules.
<value>	Minislot size for the specified upstream port. Valid values: 2, 4, 8, 16, and 32.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 minislot-size 8
cli:172.16.19.10:interface:cable:csi(1/1/1)# show upstream 1 | include "Slot Size"
Slot Size                                   8  (uSec)
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421
- **show upstream** on page 547

upstream <port number> modulation-profile

Specifies the modulation profile to be used by this specified upstream port.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
upstream <port number> modulation-profile <profile index>
```

Arguments

<port number>	Number of upstream port to which you want to apply the specified modulation profile.
<profile index>	Number of the modulation profile that you want to apply to the specified port.

Example

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1  modulation-profile 1
cli:172.16.19.10:interface:cable:csi(1/1/1)# show upstream 1 | include Modulation
Modulation Profile                          1
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421
- **show upstream** on page 547

upstream <port number> no shutdown

Sets the channel status of the specified upstream port to up.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

upstream <port number> **no shutdown**

Arguments

<port number>	Upstream port number for which you want to configure status. Valid range is 1 to 4 for 1x4 modules and 1 to 6 for 1x6 modules.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# show upstream 1 | include Status
Admin Status                              down
Operational Status                        down
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 no shutdown
cli:172.16.19.10:interface:cable:csi(1/1/1)# show upstream 1 | include Status
Admin Status                                up
Operational Status                          up
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421
- **show upstream** on page 547

upstream <port number> power-level

The receive power level for the upstream interface in Tenth dBmV.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
upstream <port number> power-level <power level>
```

Arguments

<port number>	Number of the upstream port (1-4) for which you want to set the power level.
<power level>	Power level for this upstream port. Valid range: -160 to 260 Tenth dBmV

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 power-level 0
cli:172.16.19.10:interface:cable:csi(1/1/1)# show upstream 1 | include Receive
Receive Power                              0  (1/10 dBmV)
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421
- **show upstream** on page 547

upstream <port number> range-backoff

Sets the range backoff for the specified upstream port.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

upstream <port number> range-backoff <start range> <end range>

Arguments

<port number>	Number of the upstream port (1-4) for which you want to set the power level.
<start range>	Start of range backoff for this upstream port. Valid numbers: 0 – 15.
<end range>	End of range backoff for this upstream port. Valid numbers: 0 – 15.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 range-backoff 0 15
cli:172.16.19.10:interface:cable:csi(1/1/1)# show upstream 1 | include Ranging
Ranging Backoff Start                    0
Ranging Backoff End                     15
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421
- **show upstream** on page 547

upstream <port number> ranging

Configures ranging on the specified upstream port. This allows you to tune how cable modems adjust power levels during the ranging process.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
upstream <port number> ranging {init-range-timeout <number> |
  max-power-adjust <number> | power-offset-threshold <number> |
  zero-frequency-adjust {disable|enable} | zero-power-adjust
  {disable|enable}| zero-timing-adjust {disable|enable}}
```

Arguments

<port number>	Number of the upstream port for which you want to configure ranging parameters.
init-range-timeout <number>	This is the minimum time allowed for a cable modem following receipt of a RNG-RSP, before it is expected to reply to an invitation to range request in milliseconds. By default, the cm range invite timeout is set at 400 milliseconds.
max-power-adjust <number>	The maximum adjustment permitted on a single Range Response message. By default, maximum power adjustment is set at 6 dB.
power-offset-threshold <number>	If power level offset reported by MAC chip is less than or equal to this threshold value, then power level adjustment may be stopped. By default, power offset threshold is set at 8. Values range from 0 to 4294967295.
zero-frequency-adjust {disable | enable}	If enabled, the frequency adjustment item in range response message is unconditionally set to 0. Useful for debugging. By default, zero frequency adjustment is disabled.
zero-power-adjust {disable | enable}	If enabled, the power adjustment field in the range response message is unconditionally set to 0. Useful for debugging. By default, zero power adjustment is disabled.
zero-timing-adjust {disable | enable}	If enabled, the timing adjustment item in range response message is unconditionally set to 0. Useful for debugging. By default, timing adjustment is disabled.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 ranging power-offset-th
reshold 8
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 ranging init-range-time
out 400
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 ranging max-power-adjus
t 6
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 ranging zero-power-adju
st disable
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 ranging zero-timing-adj
ust disable
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 ranging zero-frequency-
adjust disable
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421
- **show upstream** on page 547

upstream <port number> shutdown

Sets the status of the specified upstream port to down.

Mode

`interface:cable:csi(<c/s/i>)`

Syntax

`upstream` `<port number>` **`shutdown`**

Arguments

<port number>	Number of upstream port (1-4 for 1x4 modules or 1-6 for 1x6 modules) for which you want to configure status.

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# show upstream 1 | include Status
Admin Status                              up
Operational Status                        up
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 shutdown
cli:172.16.19.10:interface:cable:csi(1/1/1)# show upstream 1 | include Status
Admin Status                            down
Operational Status                      down
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Related Commands

- **show interface cable <c/s/i> mac** on page 421
- **show upstream** on page 547

upstream <port number> voice-bw-reserve

Reserves a percentage of upstream bandwidth for voice traffic on the current interface. If the admission control function is enabled on the current interface, the **upstream <port number> voice-bw-reserve** command sets aside bandwidth for use by unsolicited grant service (UGS) and UGS with activity detection (UGS/AD) service flows. Refer to the description of the **admission-control** command on page 51 for information on enabling the admission control function.

Mode

```
interface:cable:csi(<c/s/i>)
```

Syntax

```
upstream <port number> voice-bw-reserve <number>
```

Arguments

<port number>	Number of upstream port (1-4 for 1x4 modules or 1-6 for 1x6 modules) for which you want to set aside a percentage of bandwidth for UGS and UGS/AD service flows.
voice-bw-reserve <number>	Specifies the percentage of upstream bandwidth that you want to set aside for UGS and UGS/AD service flows. Values range from 0.0 to 100.0. The default is 75.0 percent. You may specify a whole number (for example, 60) or a decimal number (for example, 70.5).

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# upstream 1 voice-bw-reserve 65
cli:192.168.208.3:interface:cable:csi(1/1/1)# show upstream 1
Upstream Channel Id        1 (1 / 1 / 3 / 2)
Center Frequency                      20.0  (MHz)
Channel Width                       3200.0  (KHz)
Slot Size                                2  (uSec)
Receive Power                            0  (TenthdBmV)
Voice BW Reservation                  65.0  (%)
Modulation Profile                       1
Tx Timing Offset                         0
Tx Backoff Start                         5
Tx Backoff End                          10
Ranging Backoff Start                    2
Ranging Backoff End                      3

Admin Status                            up
Operational Status                      up

Stats:
In Octets                          2752130
In Unicast Packets                    8540
In Multicast Packets                     0
In Broadcast Packets                  8468
In Errors                                0
In Discards                              0
--More--
```

Related Commands

- **admission-control** on page 51
- **show upstream** on page 547

3 ACCOUNT ADMINISTRATION COMMANDS

This section contains an alphabetical listing of CLI commands that you can use to configure user accounts and associated access profile parameters. The following information is provided for each command:

- Description
- Mode
- Syntax
- Arguments
- Example
- Related Commands

access-profile

Creates access profiles. Access profiles are applied to user accounts to define the functional areas accessible to the user and the access rights (read/write) for those areas.

An access profile can define *read*, *write*, or *no access* privileges to the following functional areas:

- **Admin —** Functions associated with administering user accounts, such as adding modifying, and deleting users and profiles.
- **HFC** — Functions associated with configuring and monitoring DOCSIS-related (CMTS) parameters such as configuring upstream and downstream channels.
- **Observer** — Functions associated with a limited command set. The user has access to root mode and slot mode only, and is restricted to a limited number of commands. The user can type **help** or **?** to determine the available command set.
- **Prov** — Functions associated with provisioning-related tasks, such as configuring DHCP servers and subnets.
- **Router** — Functions associated with router-related tasks, such as configuring IP, RIP and OSPF interfaces.

You can apply access profiles to specific user accounts using the **account** command; you can remove then as well. Using the access profile command, you can create a pool of access profiles to select from when creating user accounts. You can apply multiple profiles for a single account. In the event of access and privilege overlap, the more powerful privileges take precedence.



Default access profiles are indicated in capital letters when viewed with the show access-profile command.

Mode

```
root
```

Syntax

```
access-profile <profile name> description <text string>
  {addprivilege | removeprivilege} {admin | hfc | observer | prov |
  router} {noaccess | read/write | readonly}
```

Arguments

<profile name>	Name that you want to assign to the profile. You can enter up to 256 alphanumeric characters. Names that contain spaces must be enclosed in quotes.
description <text string>	Name of the access-profile. You can enter up to 256 alphanumeric characters. Strings that contain spaces must be enclosed in quotes.
addprivilege	Use this keyword to add a privilege to the access profile. For example, use this command to add admin/read `only` privileges.
removeprivilege	Use this keyword to remove a privilege from the access profile. For example, use this command to remove `admin/read only` privileges.
admin	Keyword that specifies the account and chassis management functional area.
hfc	Keyword that specifies cable-related functional areas.
observer	Keyword that specifies functions associated with a limited command set. The user has access to root mode and slot mode only, and is restricted to a limited number of commands. The user can type **help** or **?** to determine the available command set.
prov	Keyword that specifies provisioning-related functional areas.
router	Keyword that specifies IP routing-related functional areas.
noaccess	Keyword that restricts both read and write access; denies all access to the specified functional area.
read/write	Keyword that provides both read and write access; the user can both view and configure the specified functional area.
readonly	Keyword that denies write access; the user can view but not configure the specified functional area.

Example

```
cli:172.16.19.10:root# access-profile routemonitor description "Readonly"
addprivilege router readonly
  PROFILE AFTER CREATE
    profileName: routemonitor
    profileDescription: Readonly
    PrivilegeList:
      router: readonly
'routemonitor' was successfully created
```

Related Commands

- **account** on page 621
- **no access-profile** on page 623
- **show access-profile** on page 625

account

Creates user accounts. Creating user accounts involves defining a username and password for the account and assigning access profiles to the user. Access profiles define the functional system areas the user can access and the type of access allowed for those areas.



Note that you must have Cuda 12000 administrative privileges to use this command.

Mode

`root`

Syntax

```
account <account name> {add-profile <profile name> |
  description <text string> | password <password> |
  remove-profile <profile name>}
```

Arguments

<account name>	Name of the user account that you want create. You can enter up to 256 characters.
add-profile <profile name>	Name of the access profile that you want to add or remove from the user account.
description <text string>	Text string providing an administrative description of the account. Text strings containing spaces must be enclosed with quotation marks (" ").
<password>	Password for this user account.

Example

```
cli:192.168.208.3:root# show access-profile

row count: 6

Profile         Description                       Route HFC   Prov  Admin Obser
--------------- --------------------------------- ----- ----- ----- ----- ---
AUDITORPROFILE  Profile with read only access     READ  READ  READ  NONE  READ
                rights to all functions except
                admin
NOACCESSPROFILE Profile with no access rights     NONE  NONE  NONE  NONE  NONE
OPERATORPROFILE Profile with read/write access    R/W   R/W   R/W   NONE  R/W
                rights to all functions except
                admin
ROOTPROFILE     Profile with root access rights   R/W   R/W   R/W   R/W   R/W
prov-user                                         NONE  NONE  R/W   NONE  NONE
test                                              NONE  NONE  NONE  NONE  NONE

cli:192.168.208.3:root# account User-A add-profile OPERATORPROFILE
description "Operator 1"
  ACCOUNT AFTER CREATE
    UserName: User-A
    UserDescription: Operator 1
    PROFILE LIST
      profileName: OPERATORPROFILE
      profileDescription: Profile with read/write access rights to all
functions
 except admin
      PrivilegeList:
         admin: noaccess
         observer: read/write
         router: read/write
         prov: read/write
         hfc: read/write
'User-A' was successfully created
```

Related Commands

- **access-profile** on page 618
- **no account** on page 624
- **show account** on page 627

no access-profile

Removes an access profile from the system. Note that you cannot remove the following default profiles:

- AUDITORPROFILE
- NOACCESSPROFILE
- OPERATORPROFILE
- ROOTPROFILE

Mode

`root`

Syntax

`no access-profile` `<profile name>`

Arguments

<profile name>	Specifies the name of an access profile.

Example

```
cli:172.16.19.10:root# show access-profile test
Showing single profile:
    profileName: test
    profileDescription: null
    PrivilegeList:
cli:172.16.19.10:root# no access-profile test
'test' was successfully removed
cli:172.16.19.10:root#
```

Related Commands

- **account** on page 621
- **show access-profile** on page 625

no account

Deletes the specified user account from the system.

Mode

root

Syntax

no account <account name>

Arguments

<account name>	Name of the user account that you want to remove from the system.

Example

```
cli:192.168.208.3:root# show account

row count: 2

Account Name           Profile                    Route HFC   Prov  Admin Obser
---------------------- -------------------------- ----- ----- ----- ----- ---
User-A                 OPERATORPROFILE              R/W   R/W   R/W   NONE
...................... .......................... ..... ..... ..... ..... ...
root                   ROOTPROFILE                  R/W   R/W   R/W   R/W
...................... .......................... ..... ..... ..... ... .....

cli:192.168.208.3:root# no account User-A
'User-A' was successfully removed
cli:192.168.208.3:root#
```

Related Commands

- **account** on page 621
- **show account** on page 627

show access-profile

Displays the access-profiles currently defined on the system. To show all access-profiles defined on the system, enter the command without the optional parameter.

Mode

root

Syntax

show access-profile [<profile name>]

Arguments

<profile name>	Name of the access-profile that you want to display.

Example

```
cli:192.168.208.3:root# show access-profile

row count: 6

Profile         Description                       Route HFC   Prov  Admin Obser
--------------- --------------------------------- ----- ----- ----- ----- ---
AUDITORPROFILE  Profile with read only access      READ  READ  READ  NONE  READ
                rights to all functions except
                admin
NOACCESSPROFILE Profile with no access rights      NONE  NONE  NONE  NONE  NONE
OPERATORPROFILE Profile with read/write access      R/W   R/W   R/W   NONE  R/W
                rights to all functions except
                admin
ROOTPROFILE     Profile with root access rights    R/W   R/W   R/W   R/W   R/W
prov-user                                         NONE  NONE  R/W   NONE  NONE
```

Related Commands

- **access-profile** on page 618
- **account** on page 621
- **no access-profile** on page 623

show active-user

Displays active user sessions. You can use the command to display all users currently logged into the Cuda 12000 environment.

Mode

root

Syntax

show active-user

Example

```
cli:172.16.19.10:root# show active-user
Displaying active users ...
SessionId: CLI:32a8ff8e:da1f1a9ffd:-4332     UserName: root
cli:172.16.19.10:root#
```

Related Commands

- **show user** on page 550

show account

Displays a list of user accounts currently defined on the system. You must have administrative privileges to use this command.

For each account, this command displays the account name, associated profile, and resulting access privileges.

Mode

`root`

Syntax

`show account` [<account-name>]

Arguments

<account name>	Name of the account for which you want to show information.

Example

```
cli:192.168.208.3:root# show account

row count: 2

Account Name          Profile                    Route HFC   Prov  Admin Obser
--------------------- -------------------------- ----- ----- ----- ----- ---
User-A                 OPERATORPROFILE            R/W   R/W   R/W   NONE
...................... ........................... ..... ..... ..... ..... ...
root                   ROOTPROFILE                R/W   R/W   R/W   R/W
...................... ........................... ..... ..... ..... ... .....
```

Related Commands

- **account** on page 621
- **no account** on page 624

A GLOSSARY

16 QAM

Modulation mode used by the CMTS. QAM uses both amplitude and phase modulation to encode multiple bits of data in one signaling element, thus achieving higher data transfer rates than just amplitude or phase modulation alone.

16 QAM encodes four bits per symbol as one of sixteen possible amplitude and phase combinations. 16 QAM refers to the number of discrete phase/amplitude states that are used to represent data bits.

64 QAM

A modulation mode used by the CMTS. 64 QAM uses both amplitude and phase modulation to encode multiple bits of data in one signaling element. 64 QAM encodes 6 bits per symbol as one of 64 possible amplitude and phase combinations.

256 QAM

A modulation mode used by the CMTS. 256 QAM uses both amplitude and phase modulation to encode multiple bits of data in one signaling element. 64 QAM encodes 8 bits per symbol as one of 256 possible amplitude and phase combinations.

A

Record that contains the IP address of the record's owner. Since hosts may have multiple IP addresses, multiple A records may match a given domain name.

Address Resolution Protocol (ARP)

A protocol for mapping an Internet Protocol address (IP address) to a physical machine address that is recognized in the local network. For example, in IP Version 4, the most common level of IP in use today, an address is 32 bits long. In an Ethernet local area network, however, addresses for attached devices are 48 bits long. (The physical machine address is also known as a Media Access Control or MAC address address.)

American National Standards Institute (ANSI)	The primary organization for fostering the development of technology standards in the United States.
ARP	See Address Resolution Protocol.
Bandwidth Allocation Map	The downstream MAC Management Message that the CMTS uses to allocate transmission opportunities to CMs.
Baseline Privacy Interface	Provides data privacy for DOCSIS 1.0 CMs and CMTS. BPI+, provides privacy for DOCSIS 1.1 CMs and CMTS.
BDU	See Bridge Protocol Data Unit.
Bootstrap Protocol (BOOTP)	A protocol that lets a network user be automatically configured (receive an IP address) and have an operating system boot or initiated without user involvement. The BOOTP server, managed by a network administrator, automatically assigns the IP address from a pool of addresses for a certain duration of time.
BPI	See Baseline Privacy Interface.
Bridge Protocol Data Unit (BDU)	Spanning tree protocol messages as defined in [ISO/IEC10038].
Broadband	Network technology that multiplexes multiple, independent network carriers onto a single cable or fiber. The technology is used to carry voice, video, and data over the same cable or fiber.
Broadcast	Transmission to two or more devices at the same time, such as over a bus-type local network or by satellite; protocol mechanism that supports group and universal addressing.
Broadcast Addresses	A predefined destination address that denotes the set of all data network service access points.
Cable Modem (CM)	A modulator-demodulator at subscriber locations intended for use in conveying data communications on a cable television system.
Cable Modem Termination System (CMTS)	A device located at the cable system head-end or distribution hub, that interfaces the HFC network to local or remote IP networks.

Cable Modem Termination System - Network Side Interface (CMTS-NSI)	The interface, defined in [DOCSIS3], between a CMTS and the equipment on its network side.
Cable Modem to CPE Interface (CMCI)	The interface, defined in [DOCSIS4], between a CM and CPE.
Carrier Hum Modulation	The peak-to-peak magnitude of the amplitude distortion relative to the RF carrier signal level due to the fundamental and low-order harmonics of the power-supply frequency.
Carrier-to-Noise Ratio (C/N or CNR)	The voltage difference between the digitally-modulated RF carrier and the continuous random noise. CNR is measured in decibels (dB).
CM	See Cable Modem.
CMCI	See Cable Modem to CPE Interface.
CMTS	See Cable Modem Termination System.
C/N or CNR	See Carrier-to-Noise Ratio.
CNAME	A record that contains an alias or nickname for the official domain name (also known as the canonical name).
Cross-Modulation	A form of television signal distortion where modulation from one or more television channels is imposed on another channel or channels.
Customer Premises Equipment (CPE)	Equipment at the end user's premises. This equipment may be provided by the end user or the service provider.
Data Link Layer	Layer 2 in the Open System Interconnection (OSI) architecture; the layer that provides services to transfer data over the transmission link between open systems.
DHCP	See Dynamic Host Configuration Protocol.
Distribution Hub	A location in a cable television network which performs the functions of a head-end for customers in its immediate area, and which receives some or all

	of its television program material from a Master Head-end in the same metropolitan or regional area.
DNS	See Domain Name System.
DOCSIS	Data Over Cable Service Interface Specification, developed by CableLabs. Defines interface standards for cable modems transmission and supporting equipment.
Domain Name System (DNS)	An on-line, distributed database used to map human-readable machine names into IP address for resolving machine names to IP addresses.
Downstream	The direction of data flow from the head-end (CMTS) to the subscriber (CM).
Drop Cable	Coaxial cable that connects to a residence or service location from a directional coupler (tap) on the nearest coaxial feeder cable.
Dynamic Host Configuration Protocol (DHCP)	A protocol that allows dynamic assignment of IP addresses to CPEs. DHCP is also used to assign IP addresses to CMs.
Dynamic Range	The ratio between the greatest signal power that can be transmitted over a multichannel analog transmission system without exceeding distortion or other performance limits, and the least signal power that can be utilized without exceeding noise, error rate or other performance limits.
Ethernet	A networking standard running speeds of 1 Gbps (Gigabit Ethernet), 10 Mbps (10BaseT) or 100 Mbps (100BaseT). Ethernet typically uses twisted pair wiring or optical fiber.
EuroDOCSIS	European Data Over Cable Service Interface Specification, developed by tComLabs and CableLabs. Defines interface standards for cable modems transmission and supporting equipment.
Extended Subsplit	A frequency division scheme that allows bidirectional traffic on a single coaxial cable. In the U.S., reverse path signals come to the head-end from 5 to 42 MHz. Forward path signals go from the head-end from 50 or 54 MHz to the upper frequency limit.
FDDI	See Fiber Distributed Data Interface.
FEC	See Forward Error Correction.

Feeder Cable	Coaxial cables that run along streets within the served area and connect between the individual taps which serve the customer drops.
Fiber Node	The interface between a fiber trunk and the coaxial distribution. Fiber nodes are located in a subscribers neighborhood.
File Transfer Protocol (FTP)	A protocol that allows users to log into a remote system, identify themselves, list remote directories, and copy files to and from the remote machine. FTP understands a few basic file formats. It is more complex than Telnet in that it maintains separate TCP connections for control and data transfer.
Flow	A unidirectional data path between a cable modem and a CMTS.
Forward Error Correction (FEC)	A technique for correcting errors incurred in transmission over a communications channel by the receiver, without requiring the retransmission of any information by the transmitter; typically it involves a convolution of the transmitted bits and the appending of extra bits, using a common algorithm by both the receiver and transmitter.
FTP	See File Transfer Protocol.
Gateway	A device that communicates with two protocols and translates services between them.
Graphical User Interface (GUI)	A program that displays information using graphics instead of command line text. The user can interact with a computer operating system through a series of “windows”, also known as “point and click”
Group Delay	The difference in transmission time between the highest and lowest of several frequencies through a device, circuit or system.
Guard Time	Minimum time allocated between bursts in the upstream referenced from the symbol center of the last symbol of a burst to the symbol center of the first symbol of the following burst. The guard time should be at least the duration of five symbols plus the maximum system timing error.
GUI	See Graphical User Interface.
Harmonic Related Carrier (HRC)	A method of spacing television channels on a cable television system in exact 6-MHz increments, with all carrier frequencies harmonically related to a common reference.

Head-End The central location on the cable network that originates the broadcast video and other signals in the downstream direction. See also Master Head-end, Distribution Hub.

Header Protocol control information located at the beginning of a protocol data unit.

HF See High Frequency.

HFC See Hybrid Fiber/Coaxial.

High Frequency (HF) The entire subsplit (5-30 MHz) and extended subsplit (5-42 MHz) band used in reverse channel communications over the cable television network.

High Return A frequency division scheme that allows bi-directional traffic on a single coaxial cable. Reverse channel signals propagate to the head-end above the downstream passband.

HRC See Harmonic Related Carrier.

Hum Modulation Undesired modulation of the television visual carrier by the fundamental or low-order harmonics of the power supply frequency, or other low-frequency disturbances.

Hybrid Fiber/Coaxial (HFC) System A broadband bidirectional shared-media transmission system using fiber trunks between the head-end and the fiber nodes, and coaxial distribution from the fiber nodes to the customer locations.

Hybrid Fiber/Coaxial (HFC) Network A network where the trunk of the cable plant is fiber technology. The fiber is connected to a coaxial cable and the signal is converted so that it is compatible to that media. The coaxial cable runs through the branches of the network and is dropped into the subscriber's home.

ICMP See Internet Control Message Protocol.

IEEE See Institute of Electrical and Electronic Engineers.

IETF See Internet Engineering Task Force.

IGMP See Internet Group Management Protocol.

Impulse Noise Noise characterized by non-overlapping transient disturbances.

Incremental Related Carriers (IRC)	A method of spacing NTSC television channels on a cable television system in which all channels except 5 and 6 correspond to the standard channel plan, used to reduce composite triple beat distortions.
Information Element	The fields that make up a MAP and define individual grants, deferred grants, etc.
Ingress Noise	A type of noise that is the major source of cable system noise. It is caused by discrete frequencies picked up by the cable plant from marine and radio broadcasts or from improperly grounded or shielded home appliances such as a hair dryer.
Initial Ranging	A process in which a cable modem acquires the correct timing offset so that it can accurately transmit using the correct mini-slot. Each cable modem obtains a timing offset; the timing offset depends on the time difference of the distance of the cable modem from the CMTS. Initial ranging is performed at cable modem initialization.
Institute of Electrical and Electronic Engineers (IEEE)	An organization of electrical engineers. The IEEE fosters the development of standards that often become national and international standards. Many IEEE standards are network interface standards.
International Organization for Standardization (ISO)	An international standards body, commonly known as the International Standards Organization.
International Telecommunications Union (ITU-T)	The Telecommunication Standardization Sector of the International Telecommunications Union is the primary international body for fostering cooperative standards for telecommunications equipment and systems.
Internet Control Message Protocol (ICMP)	An Internet network-layer protocol.
Internet Engineering Task Force (IETF)	A group that defines standard Internet operating protocol, such as TCP/IP.

Internet Group Management Protocol (IGMP) A network-layer protocol for managing multicast groups on the Internet. IGMP establishes and maintains a database of group multicast addresses and the interfaces to which a multicast router must forward the multicast data packets.

Internet Protocol (IP) The method or protocol by which data is sent from one computer to another on the Internet. Each computer (known as a host) on the Internet has at least one IP address that uniquely identifies it from all other computers on the Internet. An IP address includes a network address and a host address. An IP address can also be divided by using classless addressing or subnetting.

Interval Usage Code A field in MAPs and UCDs to link burst profiles to grants.

IP See Internet Protocol.

IP Filtering IP filtering enables you to filter upstream packets that pass through the CMTS. IP filtering can prevent subscribers from accessing head-end servers, enforce subscribers to log on to the cable network, enforce separately-billed service packages for data, and provide group access control for IP Multicast.

IP Multicast IP Multicast reduces traffic on a network by delivering a single stream of information to multiple users at one time.

IP Network A group of IP routers that route IP datagrams. These routers are sometimes referred to as Internet gateways. Users access the IP network from a host. Each network in the Internet includes some combination of hosts and IP routers.

IRC See Incremental Related Carriers.

ISO See International Organization for Standardization.

ITU-T See International Telecommunications Union.

Java A high level programming language developed by Sun Microsystems.

LAN See Local Area Network.

Latency The time delay, expressed in quantity of symbols, taken for a signal element to pass through a device.

Layer	A subdivision of the Open System Interconnection (OSI) architecture, constituted by subsystems of the same rank.
LDAP	See Lightweight Directory Access Protocol.
Lightweight Directory Access Protocol (LDAP)	A set of protocols for accessing information directories. LDAP is based on the standards contained within the X.500 standard, but is significantly simpler. And unlike X.500, LDAP supports TCP/IP, which is necessary for any type of Internet access to a directory server.
LLC	See Logical Link Control Procedure.
Local Area Network (LAN)	A non-public data network in which serial transmission is used for direct data communication among data stations located on the user's premises.
Logical Link Control (LLC) Procedure	In a local area network (LAN) or a Metropolitan Area Network (MAN), that part of the protocol that governs the assembling of data link layer frames and their exchange between data stations, independent of how the transmission medium is shared.
MAC	See Media Access Control.
Management Information Base (MIB)	A logical structure, used by the SNMP manager and agent, of the parameters needed for configuring, monitoring, or testing an SNMP device. The MIB is a hierarchical-naming structure used to uniquely identify SNMP objects (parameters). It is typically illustrated as an inverted tree.
Master Head-End	A head-end that collects television program material from various sources by satellite, microwave, fiber, and other means and distributes this material to Distribution Hubs in the same metropolitan or regional area. A Master Head-end MAY also perform the functions of a Distribution Hub for customers in its own immediate area.
Media Access Control (MAC) Address	A MAC address is used by the link layer protocol to forward packets "one hop at a time" between the host and the first router and between the first router and the next router and so on through the network until the packet arrives at it's final destination.
Media Access Control (MAC) Procedure	In a subnetwork, that part of the protocol that governs access to the transmission medium independent of the physical characteristics of the medium, but taking into account the topological aspects of the subnetworks, in order to enable the exchange of data between nodes. MAC

	procedures include framing, error protection, and acquiring the right to use the underlying transmission medium.
Media Access Control (MAC) Sublayer	The part of the data link layer that supports topology-dependent functions and uses the services of the Physical Layer to provide services to the logical link control (LLC) sublayer.
MIB	See Management Information Base.
Micro-reflections	Echoes in the forward transmission path due to departures from ideal amplitude and phase characteristics.
Mid Split	A frequency division scheme that allows bi-directional traffic on a single coaxial cable. Reverse channel signals propagate to the head-end from 5 to 108 MHz. Forward path signals go from the head-end from 162 MHz to the upper frequency limit. The diplex crossover band is located from 108 to 162 MHz.
Mini-Slot	A power-of-two multiple of 6.25 microsecond increments. For example, 1, 2, 4, 8, 16, 21, 64 or 128 times 6.25 microseconds. Mini-slots are used to divide the upstream bandwidth into discrete increments.
Moving Picture Experts Group (MPEG)	A group which develops standards for digital compressed moving pictures and associated audio.
MPEG	See Moving Picture Experts Group.
MSO	Multi System Operator
Multimedia Terminal Adapter (MTA)	A hardware interface between a computer and an Integrated Services Digital Network line needed for Voice Over IP.
Multipoint Access	User access in which more than one terminal equipment is supported by a single network termination.
Multipoint Connection	A connection among more than two data network terminations.

National Cable Television Association (NCTA)	A voluntary association of cable television operators which, among other things, provides guidance on measurements and objectives for cable television systems in the United States.
National Television Systems Committee (NTSC)	A committee which developed a set of standard protocol for television broadcast transmission and reception in the United States.
NCTA	See National Cable Television Association.
NEBS	See Network Equipment Building Systems.
Network Equipment Building Systems (NEBS)	NEBS is a Telcordia standard defining the physical, electrical, and environmental conditions under which network equipment must operate. NEBS includes: temperature, humidity, airborne contamination, fire resistance, earthquake and vibration, noise, electrical safety, lightning and surge immunity, ESD immunity, and electro-magnetic compatibility.
Network Layer	Layer 3 in the Open System Interconnection (OSI) architecture; the layer that establishes a path between open systems.
NS	Record that contains the domain name of the authoritative name server for the domain.
NTSC	See National Television Systems Committee.
Open Systems Interconnection (OSI)	A framework of ISO standards for communication between different systems made by different vendors, in which the communications process is organized into seven different categories that are placed in a layered sequence based on their relationship to the user. Each layer uses the layer immediately below it and provides a service to the layer above. Layers 7 through 4 deal with end-to-end communication between the message source and destination, and layers 3 through 1 deal with network functions.
Open Shortest Path First (OSPF)	An Interior Gateway Routing Protocol that use link-state algorithms to send routing information to all nodes in an OSPF area by calculating the shortest path to each node based on a map of the network constructed by each node. Each router sends that portion of the routing table (keeps track of routes to particular network destinations) that describes the state of its own links, and it also sends the complete routing structure of the network.
OSI	See Open Systems Interconnection.

OSPF	See Open Shortest Path First.
Packet Identifier (PID)	A unique integer value used to identify elementary streams of a program in a single- or multi-program MPEG-2 stream.
PHY	See Physical Layer.
Physical (PHY) Layer	Layer 1 in the Open System Interconnection (OSI) architecture. It provides services to transmit bits or groups of bits over a transmission link between open systems and which entails electrical, mechanical, and handshaking procedures.
Physical Media Dependent (PMD) Sublayer	A sublayer of the Physical Layer that transmits bits or groups of bits over particular types of transmission link between open systems. It entails electrical, mechanical, and handshaking procedures.
PID	See Packet Identifier.
PMD	See Physical Media Dependent Sublayer.
Protocol	A set of rules and formats that determines the communication behavior of layer entities in the performance of the layer functions. It may still require an authorization exchange with a policy module or external policy server prior to admission.
PTR	A record that contains a pointer to another part of the domain name space. This record is typically used in reverse zones.
QAM	See Quadrature Amplitude Modulation.
QoS	See Quality of Service.
Quadrature Amplitude Modulation (QAM)	A method of modulating digital signals onto a radio-frequency carrier signal involving both amplitude and phase coding. This achieves a higher data transfer rate than just amplitude or phase modulation alone.
Quality of Service	A networking term that specifies a guaranteed throughput level and end to end latency for traffic on the network.
Radio Frequency (RF)	Signals that are used by the CMTS transmitter and receiver to send data over HFC network. A radio frequency carrier is modulated to encode the digital data stream for transmission across the cable network.

Request For Comments (RFC)	A technical policy document of the IETF; these documents can be accessed on the World Wide Web at http://ds.internic.net/ds/rfcindex.html.
Return Loss	The parameter describing the attenuation of a guided wave signal (e.g., via a coaxial cable) returned to a source by a device or medium resulting from reflections of the signal generated by the source.
RF	See Radio Frequency.
RF DVT	Radio Frequency Design Verification Test.
RFC	See Request For Comments.
RIP	Routing Information Protocol.
Routing Information Protocol (RIP)	A routing protocol used for IP networks. The RIP protocol calculates the shortest distance between the source and destination address based on the lowest hop count.
Service Identifier (SID)	A mapping between the CM and the CMTS based on which bandwidth is allocated to the CM by the CMTS and by which COS is implemented. Within a MAC domain, all SIDs are unique.
SID	See Service Identifier.
Simple Network Management Protocol (SNMP)	A network management protocol used to monitor IP routers, other network devices, and the networks to which they attach.
SNAP	See Subnetwork Access Protocol.
SNMP	See Simple Network Management Protocol.
SOA	Start of Authority record. The purpose of the soa record is to inform other DNS servers how to treat information that the local server provides about the domain.
SOHO	Small Office Home Office
SSRAM	Synchronous Static RAM.

Subnet A network subdivided into networks or subnets. When subnetting is used, the host portion of the IP address is divided into a subnet number and a host number. Hosts and routers identify the bits used for the network and subnet number through the use of a subnet mask.

Subnet Mask A bit mask that is logically ANDed with the destination IP address of an IP packet to determine the network address. A router routes packets using the network address.

Subnetwork Access Protocol (SNAP) An extension of the LLC header to accommodate the use of 802-type networks as IP networks.

Subscriber A user in the home who accesses a data service.

Subsplit A frequency-division scheme that allows bi-directional traffic on a single cable. Reverse path signals come to the head-end from 5 to 30 (up to 42 on Extended Subsplit systems) MHz. Forward path signals go from the head-end from 50 or 54 MHz to the upper frequency limit of the cable network.

TCP See Transmission Control Protocol.

TFTP See Trivial File-Transfer Protocol.

Tick 6.25-microsecond time intervals that are the reference for upstream mini-slot definition and upstream transmission times.

Tilt Maximum difference in transmission gain of a cable television system over a given bandwidth (typically the entire forward operating frequency range).instant at which the last bit of the same PDU crosses a second designated boundary.

TLV See Type/Length/Value.

Transmission Control Protocol (TCP) A reliable stream service which operates at the transport-layer Internet protocol which ensures successful end-to-end delivery of data packets without error.

Transmission Convergence Sublayer A sublayer of the Physical Layer that provides an interface between the Data Link Layer and the PMD Sublayer.

Transmission Medium	The material on which information signals may be carried; e.g., optical fiber, coaxial cable, and twisted-wirepairs.
Transport Stream	In MPEG-2, a packet-based method of multiplexing one or more digital video and audio streams having one or more independent time bases into a single stream.
Trivial File-Transfer Protocol (TFTP)	An Internet protocol for transferring files without the requirement for user names and passwords that is typically used for automatic downloads of data and software.
Trunk Cable	Cables that carry the signal from the head-end to groups of subscribers. The cables can be either coaxial or fiber depending on the design of the system.
Type/Length/Value (TLV)	An encoding of three fields, in which the first field indicates the type of element, the second the length of the element, and the third field the value.
UCD	See Upstream Channel Descriptor.
UDP	See User Datagram Protocol.
UHF	See, Ultra-High Frequency.
Ultra-High Frequency	The range of the radio spectrum is the band extending from 300 MHz to 3 GHz. The wavelengths corresponding to these limit frequencies are 1 meter and 10 centimeters.
Upstream	The direction of the data flow from the subscriber location (CM) toward the head-end (CMTS).
Upstream Channel Descriptor (UCD)	A MAC management message transmitted by the CMTS Adapter Module at a configured period of time. A UCD defines the characteristics of an upstream channel including the size of the mini-slot, the upstream channel ID, and the downstream channel ID. It also defines channel parameters and a burst descriptor. UCDs are transmitted for each upstream channel.
User Datagram Protocol (UDP)	In conjunction with IP, UDP provides unreliable connection-less datagram delivery service. UDP can address specific protocol ports as a destination within a given host.

Very High Frequency (VHF) The range of the radio spectrum is the band extending from 30 MHz to 300 MHz. The wavelengths corresponding to these limit frequencies are 10 meters and 1 meter.

VGA Video Graphics Array display system.

VHF See Very High Frequency.

INDEX

C

command line interface
- about 21
- accessing
  - local 23
  - SSH 24
  - telnet 24
- command mode
  - global commands 28
  - IP interface 36
  - OSPF global configuration 37
  - physical interface 32
  - RIP configuration 40
  - root 30
  - slot 42
- show mode 26

commands
- aaa authentication 44
- access-class 45
- access-list 47
- access-profile 618
- account 621
- add arp 50
- admission-control 51
- alarm-throttle 52
- arp timeout 53
- asbr 54
- aux-device ac-monitor 55
- aux-device backplane-clock-a 56
- aux-device backplane-clock-b 57
- aux-device db15 alarm 58
- aux-device dc-monitor 62
- aux-device fan-rotation 63
- aux-device fan-temp 64
- aux-device ps-temp 65
- basmonitor 66
- boot 67
- bootp-policy 68
- bridge-group 70
- bridge-interface 72
- bridge-timeout 74
- ccdown 76
- chassis-config 78
- chassis-fault 80
- clear counters 84
- clock-source 87
- cm cpe-reset 104
- cm modify active 109
- cm modify cm-downstream 111
- cm modify cm-upstream 113
- cm modify cpe-downstream 115
- cm modify cpe-upstream 117
- cm modify learnable 119
- cm modify max-ip 121
- cm modify upstream 123
- cm reset 125
- cm-filter 88
- cm-filter-default cm downstream 92
- cm-filter-default cm upstream 94
- cm-filter-default cpe downstream 96
- cm-filter-default cpe upstream 98
- cm-offline clear 100
- cm-offline persist 102
- cm-offline timer 103
- connect 129
- cpe-control active 130
- cpe-control learnable 131
- cpe-control max-ip 132
- cpu-utilization 133
- crc 134
- db-check 135
- db-connect 136
- del arp 138
- dhcp-authority 139
- dhcp-policy 142
- dhcp-relay 145
- downstream frequency 148
- downstream interleave-depth 150
- downstream modulation 151
- downstream no shutdown 153
- downstream shutdown 154
- downstream transmit-power 155
- enable 158
- event-config reporting 159
- event-config syslog 162
- event-config throttle 163
- event-log clear 165

export 166
filter-aging 167
flap-list aging 169
flap-list clear 170
flap-list insertion-time 172
flap-list power-adj-threshold 174
flap-list size 175
help 176
http-server 178
import 179
insertion-interval 180
interface 181
ip address 183
ip filter 185
ip igmp 187
ip igmp proxy 189
ip ospf area-id 191
ip ospf authentication 193
ip ospf cost 196
ip ospf dead-interval 199
ip ospf hello-interval 202
ip ospf priority 205
ip ospf retransmit-interval 208
ip ospf transit-delay 211
ip rip accept default-route 214
ip rip accept host-route 216
ip rip authentication 218
ip rip cost 221
ip rip default cost 223
ip rip disable 225
ip rip enable 227
ip rip neighbor 229
ip rip poisoned-reverse 231
ip rip receive-version 233
ip rip send default-also 237
ip rip send default-only 239
ip rip send-version 235
ip rip split-horizon 241
ip route 243
ip source-route 246
link-trap 249
lookup 250
loop 247
map-list 251
map-timer 253
match 254
modulation-profile 256
mtu 258
no access-class 260
no access-list 262
no access-profile 623
no account 624
no asbr 263
no bootp-policy 264
no clock-source 265
no crc 266
no db-connect 267
no dhcp-relay 268
no enable 269
no ip address 270
no ip igmp 271
no ip igmp proxy 273
no ip rip 274
no loop 275
no modulation-profile 276
no mtu 277
no ospf area 278
no pos flag 280
no pos report 281
no pos scramble 282
no pos threshold 283
no ppp authentication 284
no ppp chap-hostname 285
no ppp chap-password 286
no ppp ipcp-report-address 287
no ppp negotiation-count 288
no ppp pap-sent-username 289
no ppp timeout 290
no ppp username 291
no proxy-arp 292
no snmp-server community 295
no snmp-server contact 293
no snmp-server context 294, 559
no snmp-server group 296
no snmp-server host 297
no snmp-server location 299, 572
no snmp-server name 300
no snmp-server user 301
no snmp-server view 302
ospf 306
ospf area 307
ospf-vi 303
override 309
periodic-ranging-interval 311
ping 312
plant-delay 314
pll-state 315
pos flag 316
pos report 317
pos scramble 320
pos threshold 321
ppp authentication 322
ppp chap-hostname 323
ppp chap-password 324
ppp ipcp-report-address 325
ppp negotiation-count 326

ppp pap-sent-username 327
ppp username 328
privacy base auth-lifetime 331
privacy base cert-trust 332
privacy base cert-validity-periods 333
privacy base tek-lifetime 334
privacy ca-cert 335
privacy cm-cert 337
privacy encryption 338
privacy multicast mac 341
privacy multicast said 339
privacy tek 342
proxy-arp 343
qos permission 344
quit 345
radius-server 346
ranging-attempts 347
report 348
reset 350
reset rip stats 351
rip 352
root 353
route-map 355
router 356
router ospf 359
router rip 360
router-id 357
save 361
set paging 362
set password 363
set prompt 364
set time 365
set timeout 366
shared-secret 367
show access-class 369
show access-list 370
show access-profile 625
show account 627
show active-user 626
show admission-control 372
show alarm-throttle 373
show arp 374
show aux-device 376
show boot 378
show bootp-policy 379
show bridge-group 381
show bridge-timeout 383
show buffers 384
show chassis 385
show chassis-config 386
show chassis-fault status 387
show clock-synchronization 388
show cm-filter 389
show cm-filter-default 390
show cm-offline 391
show controllers 392
show cpe-control 395
show cpu-utilization 396
show db-connect 398
show dhcp-authority 399
show dhcp-policy 401
show dhcp-relay 402
show downstream 403
show event-config 404
show event-log 407
show filter-aging 409
show flash 411
show host 412
show in-counters 413
show interface cable cm-filter-default 415
show interface cable cpe-control 416
show interface cable c/s/i downstream 417
show interface cable c/s/i flap-list 420
show interface cable c/s/i mac 421
show interface cable c/s/i modem 423
show interface cable c/s/i modem summary 424
show interface cable c/s/i modulation-profile 425
show interface cable c/s/i pll-state 426
show interface cable c/s/i proxy-arp 427
show interface cable c/s/i qos 428
show interface cable c/s/i sid 429
show interface cable c/s/i signal-quality 431
show interface cable c/s/i upstream 433
show interface ethernet 435
show ip 438
show ip address 443
show ip filter 444
show ip interface 445
show ip interface source-route 447
show ip ospf 448
show ip rip 450
show link-trap 452
show lookup 453
show map-list 456
show mode 458
show modem 459
show modem mac-address 467
show modem summary 478
show mta base 479
show mta ncs 481
show mta security 485
show mta server 487
show ospf 489
show out-counters 491
show pll-state 492

show privacy auth 493
show privacy base 495
show privacy ca-cert 497
show privacy cm-cert 500
show privacy multicast 504
show privacy tek 506
show proxy-arp 508
show qos 509
show radius-server 510
show relay-to-syslog 511
show rip neighbors 512
show rip stats all 513
show rip stats current 515
show route-map 517
show running-config 519
show shared-secret 521
show snmp community 523
show snmp context 525
show snmp engineID 526
show snmp group 527
show snmp host 528
show snmp notify 530
show snmp user 532
show snmp view 533
show spectrum-group 535
show syslog server 538
show system 539
show tacacs-server 540
show time 541
show timeout 542
show topology 543
show trace-log 544
show user 550
show version 551
shutdown 553
sleep 554
slot 555
snmp-server community 556
snmp-server contact 558
snmp-server context 559
snmp-server group 561
snmp-server host 564
snmp-server location 572
snmp-server name 573
snmp-server user 574
snmp-server view 576
spectrum-group 579
spectrum-group reset upstream 581
sync-interval 584
tacacs-server 585
talk 586
trace-log 587
traceroute 593
traffic-relay 545, 596
ucd-interval 598
up 599
upstream channel-width 600
upstream data-backoff 601
upstream frequency 602
upstream map 603
upstream minislot-size 605
upstream modulation-profile 607
upstream no shutdown 608
upstream power-level 609
upstream range-backoff 610
upstream ranging 611
upstream shutdown 613
upstream voice-bw-reserve 614

O

open shortest path first (OSPF)
route filter mode
export 39
import 39

R

RIP
route filter mode
export 41
import 41

T

technical support 19